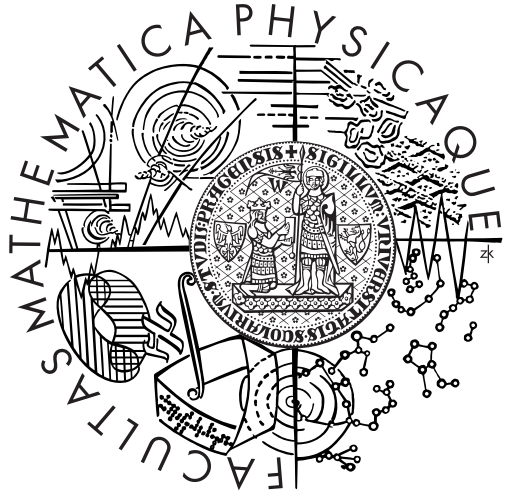
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Ria Ruppeldtová

Postranní útok na RSA-CRT s využitím SFT algoritmu

Katedra algebry

Vedoucí diplomové práce: **RNDr. Martin Hlaváč**
Studijní obor: **Matematické metody informační bezpečnosti**
Studijní program: **Matematika**

2010

Prohlašuji, že jsem svou diplomovou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 6.8.2010 Ria Ruppeldtová

Contents

Název práce: Postranní útok na RSA-CRT s využitím SFT algoritmu

Autor: Ria Ruppeldtová

Katedra: Katedra algebry

Vedoucí diplomové práce: RNDr. Martin Hlaváč

e-mail vedoucího: hlavm1am@artax.karlin.mff.cuni.cz

Abstrakt: Práce se zabývá postranním útokem na RSA s využitím Čínské věty o zbytcích a Montgomeryho násobení. Jádro útoku spočívá v hledání signifikantních koeficientů Fourierovy transformace vhodně zvolené funkce. V teoretické části je popsáno fungování SFT algoritmu a ukázány speciální vlastnosti funkce, která vychází z postranního kanálu během Montgomeryho umocňování v RSA podpisu. Na základě těchto poznatků je navrhnut nový algoritmus na hledání jednoho signifikantního koeficientu. V závěru práce jsou zkoumány vhodné vstupní parametry a je prezentována experimentálně zjištěná časová náročnost algoritmů.

Klíčová slova: RSA, Čínská věta o zbytcích, Montgomeryho umocňování, koeficienty signifikantní Fourierovy transformace

Title: Side channel attack on RSA-CRT employing SFT algorithm

Author: Ria Ruppeldtová

Department: Department of Algebra

Supervisor: RNDr. Martin Hlaváč

Supervisor's e-mail address: hlavm1am@artax.karlin.mff.cuni.cz

Abstract: The work deals with the side channel attack on RSA using Chinese Remainder Theorem and Montgomery multiplication. The core of the attack lies in finding a Significant Fourier Transform coefficient for appropriately chosen function. In the theoretical part the functionality of SFT algorithm is described and special properties of function coming from the side channel during Montgomery exponentiation in RSA signing are shown. Based on these results a new algorithm for finding a single significant coefficient is proposed. At the end of the work appropriate input parameters are explored and experimentally determined time results of algorithms are presented.

Keywords: RSA, Chinese Remainder Theorem, Montgomery exponentiation, Significant Fourier Transform coefficients

Acknowledgment

I would like to thank RNDr. Martin Hlaváč for his excellent advising, many helpful discussions and valuable suggestions. I am greatful for his friendly approach and deep scientific insight. My gratitude also belongs to RNDr. Oldřich Ulrych who willingly provided computing power for our experiments.

1 Preface

Last year the known plaintext only attack on RSA using Chinese Remainder Theorem (RSA-CRT) with Montgomery multiplication was presented in [3] by Martin Hlaváč. The special implementation of RSA allows an attacker to reveal side channel information, a number of final subtractions during Montgomery exponentiation. Hlaváč converted the retrieved information to Hidden Number Problem (HNP) and solved it by lattices, especially LLL algorithm [6]. His approach requested at least four bits of precision for the individual observations for a RSA-1024 instance.

In the same year, Adi Akavia [1] introduced an algorithm solving HNP with only one bit oracle. Her approach is based on finding significant Fourier transform (SFT) coefficients of function f from $\mathbb{Z}$ to $\mathbb{Z}_p$, i.e. elements $\alpha \in \mathbb{Z}_p$ (with p prime) such that $|\widehat{f}(\alpha)|^2 \geq \tau L_2(f)^2$, where $\widehat{f}$ is a discrete Fourier transform of the function f and $\tau \in [0, 1]$ is known. She presented a SFT algorithm over $\mathbb{Z}_p$ and later, Serge Vaudenay [12] optimized and extended it in order to work in arbitrary Abelian group $\mathbb{Z}_N$. He also proposed an algorithm for finding only a single significant Fourier coefficient.

This work builds on Hlaváč's approach using the SFT algorithm by Akavia. In our scenario, for RSA instance with $N = pq$, the side channel gives us access to the function $f(x) = \mathrm{MSMB}(xq \bmod N)$ (Most Significant Modular Bit). This oracle outputs the correct value with probability at least 95%. Since q is a significant coefficient of f, the function $\widehat{f}$ has very "nice" properties. This allows us to improve general Vaudenay's SFT algorithm. We also propose a new algorithm for finding a single SFT coefficient for this special case of the function f. The main advantage of both algorithms is that they can work with only one bit oracle and also even if the oracle to f is corrupted by "small" random noise. All of these observations lead to an adaptive chosen plaintext attack on RSA-CRT with Montgomery multiplication.

The work is organized as follows. Section 2 contains theoretical background of the SFT (mainly based on Vaudenay [12]). Especially, it focuses on the special case of the function $f(x) = \mathrm{MSMB}(xq \bmod N)$. An improved version of Vaudenay's algorithm and our new SFT algorithm are presented. In Section 3, we describe RSA-CRT with Montgomery multiplication and the side channel information which we can be acquired from this specific implementation of RSA. Finally, in Section 4, we present practical results.

2 Searching for SFT coefficients

Assume we have a function f and the goal is to find significant Fourier transform coefficients $\alpha \in \mathbb{Z}_N$ such that $|\widehat{f}(\alpha)|^2 \geq \tau L_2(f)^2$ for $\tau \in [0,1]$. By appropriate choice of the function f these results help us to factorize the number $N = pq$ for a specific instance of RSA.

2.1 Preliminaries

Convention: Throughout this work we denote the set of positive integers as $\mathbb{N}$, i.e. without considering 0 to be an element of $\mathbb{N}$.

Definition 2.1. For $a \in \mathbb{Z}$, $N \in \mathbb{N}$ we define

$$|a|_N = \min_{k \in \mathbb{Z}} |a - kN|$$

Definition 2.2. For $N \in \mathbb{N}$ we define the N-th root of unity in $\mathbb{C}$

$$\theta_N = e^{\frac{2\pi i}{N}} = \cos\frac{2\pi}{N} + i \sin\frac{2\pi}{N}$$

For ease of notation we will write θ instead of θ_N whenever it is possible.

Lemma 2.3. *Let $x, y, N \in \mathbb{N}$. Then*

$$\sum_{\alpha \in \mathbb{Z}_N} \theta^{\alpha(x-y)} = \begin{cases} 0, & \text{if } x \neq y \\ N, & \text{if } x = y \end{cases}$$

Proof. In the case $x \neq y$, the sum is given by a geometric sequence with the quotient θ^{x-y} and the first term $\theta^0 = 1$, thus

$$\sum_{\alpha \in \mathbb{Z}_N} \theta^{\alpha(x-y)} = \frac{\theta^{N(x-y)} - 1}{\theta^{(x-y)} - 1}$$

Since $\theta^N = 1$, we see that the numerator is equal to 0. On the other hand, if $x = y$, we have $\sum_{\alpha \in \mathbb{Z}_N} 1 = N$. □

Lemma 2.4. *Let $p \in \mathbb{N}$ and $m \in \mathbb{Z}_p$. Then*

$$\theta_p^{-m} = \overline{\theta_p^{-(p-m)}}$$

Proof. Applying Definition 2.2 of θ_p and the fact that $\cos(x)$ is an even and $\sin(x)$ is an odd one and both are 2π-periodic, we have

$$\begin{aligned}\overline{\theta_p^{-(p-m)}} &= \overline{e^{\frac{-2\pi i(p-m)}{p}}} = \cos\left(\frac{-2\pi(p-m)}{p}\right) - i\sin\left(\frac{-2\pi(p-m)}{p}\right) \\ &= \cos\left(-2\pi + \frac{2\pi m}{p}\right) - i\sin\left(-2\pi + \frac{2\pi m}{p}\right) \\ &= \cos\left(-\frac{2\pi m}{p}\right) + i\sin\left(-\frac{2\pi m}{p}\right) = \theta_p^{-m}\end{aligned}$$

□

Lemma 2.5. *Let $p \in \mathbb{N}$ be an odd number. Then*

$$\sum_{m=\lceil p/2\rceil}^{p-1} \theta_p^{-m} = \sum_{m=1}^{\lfloor p/2\rfloor} \overline{\theta_p^{-m}}$$

Proof.

$$\sum_{m=\lceil p/2\rceil}^{p-1} \theta_p^{-m} = \sum_{m=1}^{\lfloor p/2\rfloor} \theta_p^{-(p-m)} \overset{\text{Lemma 2.4}}{=} \sum_{m=1}^{\lfloor p/2\rfloor} \overline{\theta_p^{-m}}$$

□

Definition 2.6. For $N \in \mathbb{N}$, $\alpha \in \mathbb{Z}_N$ and complex functions f, g over $\mathbb{Z}_N$, we define

- *Discrete Fourier Transform*

$$\widehat{f}(\alpha) = \frac{1}{N}\sum_{x\in\mathbb{Z}_N} f(x)\theta^{-\alpha x}$$

- *Convolution*

$$(f \otimes g)(x) = \frac{1}{N}\sum_{y\in\mathbb{Z}_N} g(y)f(x-y)$$

- *L_1 and L_2 norms of a complex function over $\mathbb{Z}_N$*

$$L_1(f) = \sum_{x\in\mathbb{Z}_N} |f(x)|$$

$$L_2(f) = \sqrt{\frac{1}{N}\sum_{x\in\mathbb{Z}_N} |f(x)|^2}$$

Lemma 2.7 (Fourier transform decomposition)**.** *Let $N \in \mathbb{N}$ and f be a complex function over $\mathbb{Z}_N$. Then*

$$f(x) = \sum_{\alpha \in \mathbb{Z}_N} \widehat{f}(\alpha)\theta^{\alpha x}$$

Proof. By the definition of $\widehat{f}(\alpha)$, we can write

$$\begin{aligned} \sum_{\alpha \in \mathbb{Z}_N} \widehat{f}(\alpha)\theta^{\alpha x} &= \sum_{\alpha \in \mathbb{Z}_N} \left(\frac{1}{N} \sum_{y \in \mathbb{Z}_N} f(y)\theta^{-\alpha y} \right) \theta^{\alpha x} \\ &= \frac{1}{N} \sum_{y \in \mathbb{Z}_N} f(y) \sum_{\alpha \in \mathbb{Z}_N} \theta^{\alpha(x-y)} \end{aligned}$$

and we apply $\sum_{\alpha \in \mathbb{Z}_N} \theta^{\alpha(x-y)} = \begin{cases} 0, & \text{if } x \neq y \\ N, & \text{if } x = y \end{cases}$ from Lemma 2.3, thus

$$\sum_{\alpha \in \mathbb{Z}_N} \widehat{f}(\alpha)\theta^{\alpha x} = \frac{1}{N} N f(x) = f(x)$$

□

Lemma 2.8 (Parseval identity)**.** *Let $N \in \mathbb{N}$ and f be a complex function over $\mathbb{Z}_N$. Then*

$$\sum_{\alpha \in \mathbb{Z}_N} |\widehat{f}(\alpha)|^2 = L_2(f)^2$$

Proof. See [2]. □

Lemma 2.9. *Let $N \in \mathbb{N}$, $\alpha \in \mathbb{Z}_N$ and f, g be complex functions over $\mathbb{Z}_N$. Then*

$$\widehat{f \otimes g}(\alpha) = \widehat{f}(\alpha)\, \widehat{g}(\alpha)$$

Proof. See [2]. □

Theorem 2.10 (Chernoff-Hoeffding bound)**.** *Let $X_1, \ldots, X_n$ be i.i.d. complex random variables with a norm bounded by 1 and expected value μ. Then for any γ*

$$Pr\left[\left| \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \right| \geq \gamma \right] \leq 2e^{-2n\gamma^2}$$

Proof. See [4]. □

Definition 2.11. Complex functions f and g are asymptotically equivalent, abbreviated $f \approx g$, iff

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} = 1$$

2.2 Significant Fourier Transform

Adi Akavia presented new algorithm [1] to find significant Fourier transform coefficients of a function f over $\mathbb{Z}_p$, when given oracle access to f. Later, Serge Vaudenay optimized and extended Akavia's algorithm to work in arbitrary Abelian groups $\mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_k}$ [12]. In this section we mostly present Akavia's and Vaudenay's approach and improve some of the bounds, as well.

First, we define function S_L. We show how it looks and prove several of its properties. We denote weight of element α as $|\widehat{f}(\alpha)|^2$ and weight of an interval $I(c,\delta)$ (interval in $\mathbb{Z}_N$ which is centered around c and has length δ) as $\mathrm{w}(c,\delta) = \sum_{\alpha \in I(c,\delta)} |\widehat{f}(\alpha)|^2$.

This section leads to an estimate of interval weight. We also introduce expression $\bar{\mathrm{w}}(c,\delta)$ which is defined on $\mathbb{Z}_N$ but has "similar values" as $\mathrm{w}(c,\delta)$ on $I(c,\delta)$. Finally, we prove theorem on $\bar{\mathrm{W}}(c,\delta)$, an estimate of $\bar{\mathrm{w}}(c,\delta)$ given by Chernoff-Hoeffding.

Convention: If not stated otherwise, $p, q \in N$ denote two prime numbers and $N = pq$. We call this collection of parameters a RSA *instance* $\mathcal{I} = (p, q, N)$.

Definition 2.12. For an instance $\mathcal{I}$, $L \in \mathbb{Z}_N \setminus \{0\}$ and $\alpha \in \mathbb{Z}_N$, we define

$$S_L(\alpha) = \frac{1}{L} \sum_{x=0}^{L-1} \theta^{\alpha x}$$

Lemma 2.13. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$ and $\alpha \in \mathbb{Z}_N$. Then*

$$S_L(\alpha) = \frac{\sin \frac{\pi L \alpha}{N}}{L \sin \frac{\pi \alpha}{N}} \theta^{\frac{L-1}{2}\alpha}$$

Proof. Since $S_L(\alpha)$ is a geometric sum, we can write

$$\begin{aligned}
\frac{1}{L}\sum_{x=0}^{L-1} \theta^{\alpha x} &= \frac{1}{L} \cdot \frac{\theta^{L\alpha} - 1}{\theta^{\alpha} - 1} = \frac{\theta^{\frac{L-1}{2}\alpha}\theta^{\frac{\alpha}{2}}}{L\,\theta^{\frac{\alpha}{2}}} \cdot \frac{\theta^{\frac{L\alpha}{2}} - \theta^{\frac{-L\alpha}{2}}}{\theta^{\frac{\alpha}{2}} - \theta^{\frac{-\alpha}{2}}} \\
&= \frac{\theta^{\frac{L-1}{2}\alpha}}{L} \cdot \frac{\cos(\frac{2\pi}{N}\frac{L\alpha}{2}) + i\sin(\frac{2\pi}{N}\frac{L\alpha}{2}) - \cos(\frac{2\pi}{N}\frac{(-L\alpha)}{2}) - i\sin(\frac{2\pi}{N}\frac{(-L\alpha)}{2})}{\cos(\frac{2\pi}{N}\frac{\alpha}{2}) + i\sin(\frac{2\pi}{N}\frac{\alpha}{2}) - \cos(\frac{2\pi}{N}\frac{(-\alpha)}{2}) - i\sin(\frac{2\pi}{N}\frac{(-\alpha)}{2})} \\
&= \frac{\theta^{\frac{L-1}{2}\alpha}}{L} \cdot \frac{2i\sin\frac{\pi L\alpha}{N}}{2i\sin\frac{\pi\alpha}{N}} = \frac{\sin\frac{\pi L\alpha}{N}}{L\sin\frac{\pi\alpha}{N}}\theta^{\frac{L-1}{2}\alpha}
\end{aligned}$$

□

Lemma 2.14. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$ and $\alpha \in \mathbb{Z}_N$. Then*

$$|S_L(\alpha)|^2 = \frac{1}{L^2} \frac{1 - \cos\left(\frac{2\pi L\alpha}{N}\right)}{1 - \cos\left(\frac{2\pi\alpha}{N}\right)}$$

Proof. We have

$$\begin{aligned} \sin^2\varphi &= 1 - \cos^2\varphi = 1 - (\cos 2\varphi + \sin^2\varphi) \\ \sin^2\varphi &= \frac{1 - \cos 2\varphi}{2} \end{aligned}$$

which can be applied to $|S_L(\alpha)|^2$ (Lemma 2.13) where $\varphi = \frac{\pi L\alpha}{N}$, resp. $\varphi = \frac{\pi\alpha}{N}$

$$|S_L(\alpha)|^2 = \frac{\sin^2\left(\frac{\pi L\alpha}{N}\right)}{L^2 \sin^2\left(\frac{\pi\alpha}{N}\right)} = \frac{1}{L^2} \frac{1 - \cos\left(\frac{2\pi L\alpha}{N}\right)}{1 - \cos\left(\frac{2\pi\alpha}{N}\right)}$$

□

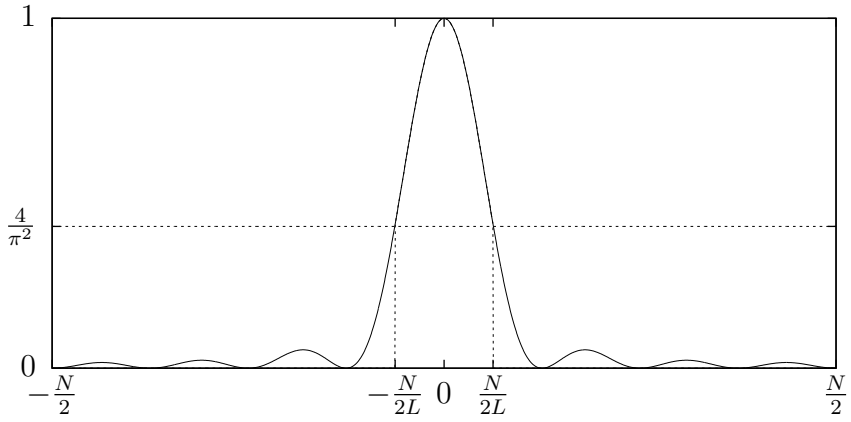


Figure 1: The function $|S_L(\alpha)|^2$ for element $\alpha \in [-\frac{N}{2}, \frac{N}{2}]$

Corollary 2.15. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$ and $\alpha \in \mathbb{Z}_N$. Then $|S_L(\alpha)|$ is an even function.*

Proof. Since sine is an odd function, by Lemma 2.13, we have

$$|S_L(-\alpha)| = \left|\frac{\sin\frac{\pi L(-\alpha)}{N}}{L\sin\frac{\pi(-\alpha)}{N}}\right| = \left|\frac{-\sin\frac{\pi L\alpha}{N}}{-L\sin\frac{\pi\alpha}{N}}\right| = |S_L(\alpha)|$$

□

Corollary 2.16. *The function $|S_L(\alpha)|^2$ is even by previous Corollary, as well.*

Lemma 2.17. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$ and $\alpha \in \mathbb{Z}_N$. Then*

(1) $|S_L(\alpha)| \leq 1$

(2) $|S_L(\alpha)| \leq \frac{\pi}{4} \cdot \frac{N/L}{|\alpha|_N}$

(3) $|S_L(\alpha)|^2 \geq 1 - \frac{\pi^2}{3} \cdot \left(\frac{\alpha}{N/L}\right)^2$ for $\alpha \in \mathbb{Z}_N$
and $|S_L(\alpha)|^2 \geq 1 - \frac{\pi^2}{12}$ if $|\alpha|_N \leq \frac{N}{2L}$

Proof.

(1) By the definition of $S_L(\alpha)$ is the arithmetic mean of unitary complex numbers.

(2) In this inequality we use:

a/ $\left|\sin\left(\frac{\pi x}{N}\right)\right| = \sin\left(\frac{\pi |x|_N}{N}\right)$

Proof. Both functions $\left|\sin\left(\frac{\pi x}{N}\right)\right|$ and $\sin\left(\frac{\pi |x|_N}{N}\right)$ are N-periodic, so consider $x \in [-\frac{N}{2}, \frac{N}{2}]$. For $x \in [0, \frac{N}{2}]$, we have $\left|\sin\left(\frac{\pi x}{N}\right)\right| = \sin\left(\frac{\pi x}{N}\right)$ and $|x|_N = x$, thus $\sin\left(\frac{\pi |x|_N}{N}\right) = \sin\left(\frac{\pi x}{N}\right)$. On the other hand for $x \in [-\frac{N}{2}, 0]$, we have $\left|\sin\left(\frac{\pi x}{N}\right)\right| = \sin\left(\frac{\pi(-x)}{N}\right)$ and $|x|_N = -x$, so $\sin\left(\frac{\pi |x|_N}{N}\right) = \sin\left(\frac{\pi(-x)}{N}\right)$. □

b/ $\frac{2}{\pi} x \leq \sin x$ for $x \in [0, \frac{\pi}{2}]$

Proof. The function $\frac{\sin x}{x}$ for $x \in [0, \frac{\pi}{2}]$ is decreasing, the smallest value is $\frac{2}{\pi}$ when $x = \frac{\pi}{2}$. □

c/ $\sin x \leq x$ for $x \geq 0$

d/ $|x|_N \leq \frac{N}{2}$

Finally, we have

$$|S_L(\alpha)| \overset{2.13}{=} \left|\frac{\sin\frac{\pi L\alpha}{N}}{L\sin\frac{\pi\alpha}{N}}\right| \overset{2a/}{=} \frac{\sin\left(\frac{\pi |L\alpha|_N}{N}\right)}{L\sin\left(\frac{\pi |\alpha|_N}{N}\right)} \overset{2b/,\,2c/}{\leq} \frac{\pi}{2L} \cdot \frac{|L\alpha|_N}{|\alpha|_N} \overset{2d/}{\leq} \frac{\pi}{4} \cdot \frac{N/L}{|\alpha|_N}$$

(3) By Lemma 2.14 we have

$$|S_L(\alpha)|^2 = \frac{1}{L^2}\frac{1-\cos\left(\frac{2\pi L\alpha}{N}\right)}{1-\cos\left(\frac{2\pi\alpha}{N}\right)}$$

and $\cos\varphi$, resp. $1-\cos\varphi$ can be bounded by the Taylor approximation

$$\begin{aligned} 1-\frac{\varphi^2}{2!} \leq &\quad \cos\varphi \quad &\leq 1-\frac{\varphi^2}{2!}+\frac{\varphi^4}{4!} \\ \frac{\varphi^2}{2!}-\frac{\varphi^4}{4!} \leq &\quad 1-\cos\varphi \quad &\leq \frac{\varphi^2}{2!} \end{aligned}$$

so we obtain

$$\begin{aligned} |S_L(\alpha)|^2 &\geq \frac{1}{L^2}\cdot\frac{\frac{\left(\frac{2\pi L\alpha}{N}\right)^2}{2}-\frac{\left(\frac{2\pi L\alpha}{N}\right)^4}{24}}{\frac{\left(\frac{2\pi\alpha}{N}\right)^2}{2}} = \frac{1}{L^2}\cdot\left(L^2-\frac{4\pi^2}{12}\cdot\frac{L^4\alpha^2}{N^2}\right) \\ &\geq 1-\frac{\pi^2}{3}\cdot\left(\frac{\alpha}{N/L}\right)^2 \end{aligned}$$

and for $|\alpha|_N \leq \frac{N}{2L}$ the estimate becomes

$$|S_L(\alpha)|^2 \geq 1-\frac{\pi^2}{3}\cdot\frac{L^2\cdot\frac{N^2}{4L^2}}{N^2} = 1-\frac{\pi^2}{12}$$

□

Remark 2.18. As we will see later, the bound (2) from the previous Lemma is useful for "big" $|\alpha|$ while the bound (1) is convenient for α close to 0.

Lemma 2.19. *Let $\mathcal{I}$ be an instance and $L \in \mathbb{Z}_N \setminus \{0\}$. Then*

(1) $\left|S_L(\frac{N}{4L})\right|^2 \geq \frac{8}{\pi^2}$

(2) $\left|S_L(\frac{N}{2L})\right|^2 \geq \frac{4}{\pi^2}$

Proof.

(1)

$$\left|S_L\left(\frac{N}{4L}\right)\right|^2 = \frac{\sin^2\left(\frac{\pi L\cdot\frac{N}{4L}}{N}\right)}{L^2\sin^2\left(\frac{\pi\cdot\frac{N}{4L}}{N}\right)} = \frac{\sin^2\frac{\pi}{4}}{L^2\sin^2\frac{\pi}{4L}} \geq \frac{\frac{1}{2}}{L^2\cdot\frac{\pi^2}{16L^2}} = \frac{8}{\pi^2}$$

(2)

$$\left|S_L\left(\frac{N}{2L}\right)\right|^2 = \frac{\sin^2\left(\frac{\pi L\cdot\frac{N}{2L}}{N}\right)}{L^2\sin^2\left(\frac{\pi\cdot\frac{N}{2L}}{N}\right)} = \frac{\sin^2\frac{\pi}{2}}{L^2\sin^2\frac{\pi}{2L}} \geq \frac{1}{L^2\cdot\frac{\pi^2}{4L^2}} = \frac{4}{\pi^2}$$

□

Definition 2.20. For an instance $\mathcal{I}$, $L \in \mathbb{Z}_N \setminus \{0\}$ and $c \in \mathbb{Z}_N$ we define the set

$$I(c, N/L) = \left\{\alpha \in \mathbb{Z}\,;|c-\alpha|_N \leq \frac{N}{2L}\right\}$$

Definition 2.21. For an instance $\mathcal{I}$, $L \in \mathbb{Z}_N \setminus \{0\}$ and $c \in \mathbb{Z}_N$ we define

$$\mathrm{w}(c, N/L) = \sum_{\alpha\in I(c,N/L)} |\widehat{f}(\alpha)|^2$$

Definition 2.22. For an instance $\mathcal{I}$, $L \in \mathbb{Z}_N \setminus \{0\}$ and $c \in \mathbb{Z}_N$ we define

$$\bar{\mathrm{w}}(c, N/L) = \sum_{\alpha\in\mathbb{Z}_N} |\widehat{f}(\alpha)|^2|S_L(c-\alpha)|^2$$

Lemma 2.23. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$ and $c \in \mathbb{Z}_N$. Then*

$$\bar{\mathrm{w}}(c, N/L) \geq \left(1-\frac{\pi^2}{12}\right)\mathrm{w}(c, N/L)$$

Proof. We divide the sum from the definition of $\bar{\mathrm{w}}(c, N/L)$ into two parts. Applying inequality (3) from Lemma 2.17 (in the first sum $|c-\alpha|_N \leq \frac{N}{2L}$), we get

$$\begin{aligned}
\bar{\mathrm{w}}(c, N/L) &= \sum_{\alpha\in I(c,N/L)} |\widehat{f}(\alpha)|^2|S_L(c-\alpha)|^2 + \sum_{\substack{\alpha\in\mathbb{Z}_N\\ \alpha\notin I(c,N/L)}} |\widehat{f}(\alpha)|^2|S_L(c-\alpha)|^2 \\
&\geq \sum_{\alpha\in I(c,N/L)} |\widehat{f}(\alpha)|^2\left(1-\frac{\pi^2}{12}\right) + 0 \\
&\geq \left(1-\frac{\pi^2}{12}\right)\mathrm{w}(c, N/L)
\end{aligned}$$

□

Definition 2.24. For an instance $\mathcal{I}$, $L \in \mathbb{Z}_N \setminus \{0\}$ and $c \in \mathbb{Z}_N$ we define the function

$$h_c(x) = \begin{cases} \frac{N}{L}\theta^{cx}, & \text{if } 0 \leq x < L \\ 0, & \text{otherwise} \end{cases}$$

Lemma 2.25. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$ and $\alpha, c \in \mathbb{Z}_N$. Then*

$$\widehat{h}_c(\alpha) = S_L(c-\alpha)$$

Proof. By definitions of $\widehat{h}_c(\alpha)$ and $S_L(c-\alpha)$ we obtain

$$\begin{aligned}
\widehat{h}_c(\alpha) &= \frac{1}{N}\sum_{x\in\mathbb{Z}_N} h_c(x)\theta^{-\alpha x} \\
&= \frac{1}{N}\sum_{x=0}^{L-1} \frac{N}{L}\theta^{cx}\theta^{-\alpha x} \\
&= \frac{1}{L}\sum_{x=0}^{L-1} \theta^{(c-\alpha)x} \;=\; S_L(c-\alpha)
\end{aligned}$$

□

Lemma 2.26. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$ and $c \in \mathbb{Z}_N$. Then*

$$\bar{\mathrm{w}}(c, N/L) = \frac{1}{NL^2}\sum_{x\in\mathbb{Z}_N}\sum_{y=0}^{L-1}\sum_{z=0}^{L-1} f(x-y)\overline{f(x-z)}\,\theta^{c(y-z)}$$

Proof. According to previous definitions and lemmas we can write

$$\begin{aligned}
\bar{\mathrm{w}}(c, N/L) &\overset{\text{Def } \bar{\mathrm{w}}(c,N/L)}{=} \sum_{\alpha\in\mathbb{Z}_N} |\widehat{f}(\alpha)|^2 |S_L(c-\alpha)|^2 \overset{\text{Lemma 2.25}}{=} \sum_{\alpha\in\mathbb{Z}_N} |\widehat{f}(\alpha)\widehat{h}_c(\alpha)|^2 \\
&\overset{\text{Lemma 2.9}}{=} \sum_{\alpha\in\mathbb{Z}_N} |\widehat{f\otimes h_c}(\alpha)|^2 \overset{\text{Parseval}}{=} L_2(f\otimes h_c)^2 \\
&\overset{\text{Def } L_2,\, f\otimes h_c}{=} \frac{1}{N}\sum_{x\in\mathbb{Z}_N}\left|\frac{1}{|\mathbb{Z}_N|}\sum_{y\in\mathbb{Z}_N} f(x-y)h_c(y)\right|^2 \\
&\overset{\text{Def } h_c}{=} \frac{1}{N}\sum_{x\in\mathbb{Z}_N}\left|\frac{1}{L}\sum_{y=0}^{L-1} f(x-y)\theta^{cy}\right|^2 \\
&= \frac{1}{NL^2}\sum_{x\in\mathbb{Z}_N}\sum_{y=0}^{L-1}\sum_{z=0}^{L-1} f(x-y)\overline{f(x-z)}\,\theta^{c(y-z)}
\end{aligned}$$

The last equality is from the relation $\lambda.\overline{\lambda} = |\lambda|^2$ where $\lambda = \sum_{y=0}^{L-1} f(x-y)\theta^{cy}$ and $\overline{\theta^{cz}} = \theta^{-cz}$. □

Lemma 2.27. *Let $\mathcal{I}$ be an instance and $L \in \mathbb{Z}_N$, further let $\mathcal{A}$, resp. $\mathcal{B}$ and $\mathcal{C}$ be mutually independent random variables on $\mathbb{Z}_N$, resp. $\mathbb{Z}_L$ and f be a complex function over $\mathbb{Z}_N$ with $|f(x)| \leq 1$ for all $x \in \mathbb{Z}_N$ and $c \in \mathbb{Z}_N$. Then for $\mathcal{X}_i \sim f(\mathcal{A}-\mathcal{B})\overline{f(\mathcal{A}-\mathcal{C})}\theta^{c(\mathcal{B}-\mathcal{C})}$ where $1 \leq i \leq n$, the following statements hold*

(1) $\mathcal{X}_1, \ldots, \mathcal{X}_n$ are i.i.d. complex random variables

(2) $\mathcal{X}_1, \ldots, \mathcal{X}_n$ are bounded by 1, i.e. $|\mathcal{X}_i| \leq 1$

(3) expected value μ of $\mathcal{X}_i$ is $\bar{\mathrm{w}}(c, N/L)$

(Remark that $\mathcal{X} \sim \mathcal{Y}$ means that $\mathcal{X}$ and $\mathcal{Y}$ have the same distibution.)

Proof.

(1) $\mathcal{X}_1, \ldots, \mathcal{X}_n$ are independent and since they are generated in the same way, they are also identically distributed.

(2) As $|f(x)| \leq 1$ and $|\theta^{c(\mathcal{B}_i-\mathcal{C}_i)}| = 1$, so $|\mathcal{X}_i| \leq 1$.

(3) In the discrete case the expected value is given by

$$\begin{aligned}
\mathrm{E}\mathcal{X}_i &= \sum_{\substack{x\in\mathbb{Z}_N \\ y,z\in\mathbb{Z}_L}} \Pr[\mathcal{A}=x,\mathcal{B}=y,\mathcal{C}=z] f(x-y)\overline{f(x-z)}\theta^{c(y-z)} \\
&= \sum_{\substack{x\in\mathbb{Z}_N \\ y,z\in\mathbb{Z}_L}} \Pr[\mathcal{A}=x]\Pr[\mathcal{B}=y]\Pr[\mathcal{C}=z] f(x-y)\overline{f(x-z)}\theta^{c(y-z)} \\
&= \sum_{\substack{x\in\mathbb{Z}_N \\ y,z\in\mathbb{Z}_L}} \frac{1}{NL^2} f(x-y)\overline{f(x-z)}\theta^{c(y-z)} \overset{2.26}{=} \bar{\mathrm{w}}(c, N/L)
\end{aligned}$$

□

Theorem 2.28. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$, $c \in \mathbb{Z}_N$, $\gamma \in \mathbb{R}$ and f be a complex function over $\mathbb{Z}_N$ with $|f(x)| \leq 1$ for all $x \in \mathbb{Z}_N$. Furthermore, let*

$$\bar{\mathrm{W}}(c, N/L) = \mathrm{Re}\left(\frac{1}{n}\sum_{i=1}^{n} f(A_i - B_i)\overline{f(A_i - C_i)}\,\theta^{c(B_i-C_i)}\right)$$

be an estimate of $\bar{\mathrm{w}}(c, N/L)$ where A resp. B, C are random vectors of length $n = \frac{1}{2}\gamma^{-2}\ln\frac{1}{\varepsilon}$ on $\mathbb{Z}_N$ resp. $\mathbb{Z}_L$. Then the estimate $\bar{\mathrm{W}}(c, N/L)$ is "good", i.e.

$$\Pr[|\,\bar{\mathrm{W}}(c, N/L) - \bar{\mathrm{w}}(c, N/L)| \geq \gamma] \leq 2\varepsilon$$

Proof. By Lemma 2.27 (2), we have $\left|f(A_i - B_i)\overline{f(A_i - C_i)}\,\theta^{c(B_i - C_i)}\right| \leq 1$ for all $i \in [1, n]$, as well. Therefore, we can apply Theorem 2.10 by Chernoff-Hoeffding. We see that $\bar{\mathrm{w}}(c, N/L)$ is an expected value for $\bar{\mathrm{W}}(c, N/L)$ by Lemma 2.27 (3), thus

$$\Pr[|\,\bar{\mathrm{W}}(c, N/L) - \bar{\mathrm{w}}(c, N/L)| \geq \gamma] \leq 2e^{-2n\gamma^2} = 2\varepsilon$$

□

2.3 Special case of the function f

Recall that we are trying to find the factor q of $N = pq$ in a particular instance of RSA. In this section we will consider functions $g(x) = xq \bmod N$, $f'(x) = \mathrm{MSMB}(g(x))$ as the Most Significant Modular Bit of $g(x)$, and also the function $f(x) = f'(x) - 1/2$. The function f', as shown later, emerges as a side channel from an attack on specific implementation of the RSA algorithm. We focus on values of the function f' which will help us to find the factor q of N.

In order to find q we will use Significant Fourier Transform (SFT) algorithm. This algorithm for given $\tau \in [0, 1]$ outputs all τ-significant coefficients α of the function f', i.e. all $\alpha \in \mathbb{Z}_N$ such that $|\widehat{f}(\alpha)|^2 \geq \tau L_2(f)^2$. In this section we prove (besides some technical lemmas) a few properties of the functions f' and f, especially we compute their L_2 norms and their discrete Fourier transforms in particular points. We will use these results to prove that the weight $|\hat{f}(kq)|^2$ of odd multiples of q is significantly bigger than the weight of other points $\alpha \in \mathbb{Z}_N$ and the weight of q is the biggest one. This result theoretically justifies the validity of the SFT algorithm.

We have decided to choose a function f instead of f' in SFT algorithm. As we show, for the function f', point 0 has big weight (actually $\frac{1}{2}L_2(f')^2$) unlike the function f where 0 has weight 0 in limit case for $p \to \infty$. In searching for significant coefficient we are not interested in finding 0, but q.

Definition 2.29 (Most Significant Modular Bits)**.** For an instance $\mathcal{I}$, $x \in \mathbb{Z}_N$ and $l \in \mathbb{N}$, we define the function

$$\mathrm{MSMB}_{l,N}(x) = \left\lfloor \frac{x}{N/2^l} \right\rfloor$$

For $l = 1$, we will write $\mathrm{MSMB}(x)$ instead of $\mathrm{MSMB}_{1,N}(x)$.

Observation 2.30. *Let $\mathcal{I}$ be an instance. Then we can write*

(a) for $x \in \mathbb{Z}_N$ $\qquad \mathrm{MSMB}_{1,N}(x) = \begin{cases} 0, & \text{if } x < N/2 \\ 1, & \text{otherwise} \end{cases}$

(b) for $x \in \mathbb{Z}_p$ $\qquad \mathrm{MSMB}_{1,N}(xq) = \begin{cases} 0, & \text{if } x < p/2 \\ 1, & \text{otherwise} \end{cases}$

Definition 2.31. For an instance $\mathcal{I}$, we define functions $g : \mathbb{Z} \to \mathbb{Z}_N$, $f' : \mathbb{Z} \to \{0, 1\}$ and $f : \mathbb{Z} \to \{-1/2, 1/2\}$ as follows

$$\begin{aligned} g(x) &= xq \ \mathbf{mod}\ N \\ f'(x) &= \mathrm{MSMB}(g(x)) \\ f(x) &= \mathrm{MSMB}(g(x)) - 1/2 \end{aligned}$$

Lemma 2.32. *Let $\mathcal{I}$ be an instance. Then the function g is p-periodic on $\mathbb{Z}$.*

Proof.

$$g(x+p) = ((x+p)\, q) \ \mathbf{mod}\ N = (xq + pq) \ \mathbf{mod}\ N = xq \ \mathbf{mod}\ N = g(x)$$

□

Corollary 2.33. *Let $\mathcal{I}$ be an instance. Then functions f' and f are p-periodic on $\mathbb{Z}$, as well.*

Lemma 2.34. *Let $\mathcal{I}$ be an instance, $\alpha \in \mathbb{Z}_N$ and F be p-periodic function. Then*

$$\widehat{F}(\alpha) = \frac{1}{N} \sum_{j=0}^{q-1} \sum_{l=0}^{p-1} F(l) \theta_q^{-\alpha j} \theta_N^{-\alpha l}$$

Proof. First, by the Definition 2.6 of Discrete Fourier Transform, we have

$$\widehat{F}(\alpha) = \frac{1}{N} \sum_{x \in \mathbb{Z}_N} F(x) \theta_N^{-\alpha x}$$

Since F is p-periodic and $N = pq$, we can write

$$\begin{aligned} \widehat{F}(\alpha) &= \frac{1}{N} \sum_{j=0}^{q-1} \sum_{l=0}^{p-1} F(jp+l) \theta_N^{-\alpha(jp+l)} = \frac{1}{N} \sum_{j=0}^{q-1} \sum_{l=0}^{p-1} F(l) \theta_N^{-\alpha(jp+l)} \\ &= \frac{1}{N} \sum_{j=0}^{q-1} \sum_{l=0}^{p-1} F(l) \theta_N^{-\alpha jp} \theta_N^{-\alpha l} = \frac{1}{N} \sum_{j=0}^{q-1} \sum_{l=0}^{p-1} F(l) \theta_q^{-\alpha j} \theta_N^{-\alpha l} \end{aligned}$$

□

Lemma 2.35. *It holds*
(a) $\lim_{p \to \infty} L_2(f')^2 = \frac{1}{2}$
(b) $L_2(f)^2 = \frac{1}{4}$

Proof. Since both f' and f are p-periodic (Corollary 2.33), we can write

(a)

$$\begin{aligned} L_2(f')^2 &= \frac{1}{N}\sum_{x\in\mathbb{Z}_N}|f'(x)|^2 = \frac{1}{N}\sum_{x\in\mathbb{Z}_N}|\mathrm{MSMB}(xq \bmod N)|^2 \\ &\overset{2.33}{=} \frac{1}{N}\sum_{j=0}^{q-1}\sum_{l=0}^{p-1}|\mathrm{MSMB}(lq \bmod N)|^2 \\ &= \frac{q}{N}\sum_{l=0}^{p-1}\mathrm{MSMB}(lq) \overset{2.30}{=} \frac{1}{p}\sum_{l=\lceil\frac{p}{2}\rceil}^{p-1}1 = \frac{1}{2}-\frac{1}{2p} \end{aligned}$$

Finally,

$$\lim_{p\to\infty} L_2(f')^2 = \frac{1}{2}$$

(b)

$$\begin{aligned} L_2(f)^2 &= \frac{1}{N}\sum_{x\in\mathbb{Z}_N}|f(x)|^2 = \frac{1}{N}\sum_{x\in\mathbb{Z}_N}|\mathrm{MSMB}(xq \bmod N)-1/2|^2 \\ &\overset{2.33}{=} \frac{1}{N}\sum_{j=0}^{q-1}\sum_{l=0}^{p-1}|\mathrm{MSMB}(lq \bmod N)-1/2|^2 \\ &= \frac{q}{N}\sum_{l=0}^{p-1}|\mathrm{MSMB}(lq)-1/2|^2 \\ &\overset{2.30}{=} \frac{1}{p}\left(\sum_{l=0}^{\lfloor p/2\rfloor}\left|-\frac{1}{2}\right|^2+\sum_{l=\lceil p/2\rceil}^{p-1}\left|\frac{1}{2}\right|^2\right) = \frac{1}{p}\sum_{l=0}^{p-1}\frac{1}{4} = \frac{1}{4} \end{aligned}$$

□

Lemma 2.36. *Let $\mathcal{I}$ be an instance. Then*

(a) $|\widehat{f'}(0)| = \frac{1}{2}-\frac{1}{2p}$

(b) $|\widehat{f}(0)| = \frac{1}{2p}$

Proof. Using the result from technical Lemma 2.34 for $\alpha = 0$ and applying Observation 2.30 and $\lfloor p/2\rfloor = \frac{p-1}{2}$, resp. $\lceil p/2\rceil = \frac{p+1}{2}$, we obtain

(a)

$$\begin{aligned}
|\widehat{f'}(0)| &= \left|\frac{1}{N}\sum_{x\in\mathbb{Z}_N} f'(x)\theta^{0x}\right| \overset{2.34}{=} \left|\frac{1}{N}\sum_{j=0}^{q-1}\sum_{l=0}^{p-1} f'(l)\right| \\
&= \left|\frac{q}{N}\sum_{l=0}^{p-1}\mathrm{MSMB}(lq \bmod N)\right| \overset{lq<N}{=} \left|\frac{1}{p}\sum_{l=0}^{p-1}\mathrm{MSMB}(lq)\right| \\
&\overset{2.30(b)}{=} \left|\frac{1}{p}\sum_{l=\lceil p/2\rceil}^{p-1} 1\right| = \frac{1}{2}-\frac{1}{2p}
\end{aligned}$$

(b)

$$\begin{aligned}
|\widehat{f}(0)| &= \left|\frac{1}{N}\sum_{x\in\mathbb{Z}_N} f(x)\theta^{0x}\right| \overset{2.34}{=} \left|\frac{1}{N}\sum_{j=0}^{q-1}\sum_{l=0}^{p-1} f(l)\right| \\
&= \left|\frac{q}{N}\sum_{l=0}^{p-1}\mathrm{MSMB}(lq \bmod N) - 1/2\right| \\
&\overset{lq<N}{=} \left|\frac{1}{p}\sum_{l=0}^{p-1}\mathrm{MSMB}(lq) - 1/2\right| \\
&\overset{2.30(b)}{=} \frac{1}{p}\left|-\frac{1}{2}\sum_{l=0}^{\lfloor p/2\rfloor} 1 + \frac{1}{2}\sum_{l=\lceil p/2\rceil}^{p-1} 1\right| = \frac{1}{2p}
\end{aligned}$$

□

Corollary 2.37. *Let $\mathcal{I}$ be an instance. Then*

(a) $\lim_{p\to\infty} |\widehat{f'}(0)|^2 = \frac{1}{4}$

(b) $\lim_{p\to\infty} |\widehat{f}(0)|^2 = 0$

Lemma 2.38. *Let $\mathcal{I}$ be an instance, F be a p-periodic function and $\beta \neq kq$ for all $k \in \mathbb{N}$. Then*

$$|\widehat{F}(\beta)| = 0$$

Proof. Based on the definition of $\widehat{F}$ and Lemma 2.34, we have

$$|\widehat{F}(\beta)| \overset{2.34}{=} \left|\frac{1}{N}\sum_{j=0}^{q-1}\sum_{l=0}^{p-1} F(l)\theta_q^{-\beta j}\theta_N^{-\beta l}\right|$$

Since $\theta_q^{-\beta j}$ does not depend on l, we can take it out from the inner sum

$$|\widehat{F}(\beta)| = \left| \frac{1}{N} \sum_{j=0}^{q-1} \left(\theta_q^{-\beta j} \sum_{l=0}^{p-1} F(l) \theta_N^{-\beta l} \right) \right|$$

Similarly, $\sum_{l=0}^{p-1} f(l)\theta_N^{-\beta l}$ is independent of j, so we get

$$|\widehat{F}(\beta)| = \left| \frac{1}{N} \left(\sum_{l=0}^{p-1} F(l) \theta_N^{-\beta l} \right) \left(\sum_{j=0}^{q-1} \theta_q^{-\beta j} \right) \right|$$

Finally, we apply $\beta \neq kq$. Notice $\sum_{j=0}^{q-1} \theta_q^{-\beta j}$ is a geometric sum. Thus

$$\sum_{j=0}^{q-1} \theta_q^{-\beta j} = \frac{\theta_q^{-q\beta} - 1}{\theta_q^{-\beta} - 1} = 0$$

since $\theta_q^{-q\beta} = (\theta_q^q)^{-\beta} = 1$. □

Corollary 2.39. *Since f is p-periodic, $|\widehat{f}(\beta)| = 0$ for all $\beta \neq kq$, $k \in \mathbb{Z}$.*

Lemma 2.40. *Let $\mathcal{I}$ be an instance and $t \in \mathbb{Z} \setminus \{0\}$. Then*

$$\sum_{l=1}^{\lfloor p/2 \rfloor} e^{\frac{2\pi i t l}{p}} = e^{\frac{2\pi i t}{p}} \cdot \frac{(-1)^t \, e^{\frac{-\pi i t}{p}} - 1}{e^{\frac{2\pi i t}{p}} - 1}$$

Proof. The sum is given by a geometric sequence with the same quotient and the first term $e^{\frac{2\pi i t}{p}}$

$$\sum_{l=1}^{\lfloor p/2 \rfloor} e^{\frac{2\pi i t l}{p}} = e^{\frac{2\pi i t}{p}} \cdot \frac{e^{\frac{2\pi i t}{p} \cdot \frac{p-1}{2}} - 1}{e^{\frac{2\pi i t}{p}} - 1} = e^{\frac{2\pi i t}{p}} \cdot \frac{(-1)^t \, e^{\frac{-\pi i t}{p}} - 1}{e^{\frac{2\pi i t}{p}} - 1}$$

□

Lemma 2.41. *Let $\mathcal{I}$ be an instance and $t \in \mathbb{Z} \setminus \{0\}$. Then*

$$\frac{1}{p} \sum_{l=1}^{\lfloor p/2 \rfloor} e^{\frac{2\pi i t l}{p}} \approx \begin{cases} \frac{1}{2p}, & \text{for } t \text{ even} \\ -\frac{1}{\pi i t}, & \text{for } t \text{ odd} \end{cases}$$

Proof. We use the result from Lemma 2.40

$$\frac{1}{p}\sum_{l=1}^{\lfloor p/2\rfloor} e^{\frac{2\pi itl}{p}} = \frac{1}{p}\cdot e^{\frac{2\pi it}{p}}\cdot\frac{(-1)^t\, e^{\frac{-\pi it}{p}}-1}{e^{\frac{2\pi it}{p}}-1}$$

further well-known $\lim_{x\to 0}\frac{e^x-1}{x}=1$ for numerator and denominator when t is even and denominator when t is odd and finally, $\lim_{p\to\infty} e^{\frac{\pi it}{p}}=1$.

(1) For t even

$$\frac{1}{p}\cdot e^{\frac{2\pi it}{p}}\cdot\frac{e^{\frac{-\pi it}{p}}-1}{e^{\frac{2\pi it}{p}}-1}\approx\frac{1}{p}\cdot\frac{-\frac{\pi it}{p}}{\frac{2\pi it}{p}}=\frac{1}{2p}$$

(2) For t odd

$$\frac{1}{p}\cdot e^{\frac{2\pi it}{p}}\cdot\frac{-e^{\frac{-\pi it}{p}}-1}{e^{\frac{2\pi it}{p}}-1}\approx\frac{1}{p}\cdot\frac{-2}{\frac{2\pi it}{p}}=-\frac{1}{\pi it}$$

□

Lemma 2.42. *Let $\mathcal{I}$ be an instance and $k\in\mathbb{N}$. Then*

$$|\widehat{f}(kq)|^2\approx\begin{cases}\frac{1}{4p^2}, & \text{for } k \text{ even}\\ \frac{1}{4p^2}+\frac{1}{(k\pi)^2}, & \text{for } k \text{ odd}\end{cases}$$

Proof. First we express $|\widehat{f}(kq)|$ and by applying Lemma 2.34, we have

$$\begin{aligned}|\widehat{f}(kq)| &\overset{2.34}{=} \left|\frac{q}{N}\sum_{l=0}^{p-1} f(l)\theta_p^{-kl}\right| = \left|\frac{1}{p}\sum_{l=0}^{p-1}(\mathrm{MSMB}(lq \bmod N)-1/2)\,\theta_p^{-kl}\right|\\ &= \left|\frac{1}{p}\sum_{l=0}^{p-1}(\mathrm{MSMB}(lq)-1/2)\,\theta_p^{-kl}\right|\end{aligned}$$

Furthermore, we use Observation 2.30 (b) and Lemma 2.5

$$\begin{aligned}
|\widehat{f}(kq)| &= \frac{1}{p}\left|\sum_{l=0}^{\lfloor p/2\rfloor}\left(-\frac{1}{2}\right)\theta_p^{-kl}+\sum_{l=\lceil p/2\rceil}^{p-1}\left(\frac{1}{2}\right)\theta_p^{-kl}\right| \\
&\overset{2.5}{=} \frac{1}{p}\left|-\frac{1}{2}+\sum_{l=1}^{\lfloor p/2\rfloor}\left(-\frac{1}{2}\right)\theta_p^{-kl}+\sum_{l=1}^{\lfloor p/2\rfloor}\left(\frac{1}{2}\right)\overline{\theta_p^{-kl}}\right| \\
&= \frac{1}{p}\left|-\frac{1}{2}-\frac{1}{2}\sum_{l=1}^{\lfloor p/2\rfloor}e^{\frac{-2\pi ikl}{p}}+\frac{1}{2}\sum_{l=1}^{\lfloor p/2\rfloor}e^{\frac{2\pi ikl}{p}}\right| \\
&= \left|-\frac{1}{2p}-\frac{1}{2p}\sum_{l=1}^{\lfloor p/2\rfloor}e^{\frac{-2\pi ikl}{p}}+\frac{1}{2p}\sum_{l=1}^{\lfloor p/2\rfloor}e^{\frac{2\pi ikl}{p}}\right|
\end{aligned}$$

Now, applying Lemma 2.41 for $t=-k$ and $t=k$ we get

(1) for k even

$$|\widehat{f}(kq)|^2 \approx \left|-\frac{1}{2p}-\frac{1}{4p}+\frac{1}{4p}\right|^2=\frac{1}{4p^2}$$

(2) for k odd

$$\begin{aligned}
|\widehat{f}(kq)|^2 &\approx \left|-\frac{1}{2p}-\frac{1}{2}\cdot\frac{1}{\pi ik}+\frac{1}{2}\cdot\left(-\frac{1}{\pi ik}\right)\right|^2 \\
&= \left|-\frac{1}{2p}+\frac{i}{\pi k}\right|^2=\frac{1}{4p^2}+\frac{1}{(k\pi)^2}
\end{aligned}$$

since $-\frac{1}{i}=i$.

□

Corollary 2.43. *Let $\mathcal{I}$ be an instance and $k\in\mathbb{N}$. Then*

$$\lim_{p\to\infty}|\widehat{f}(kq)|^2=\begin{cases}0, & \text{for } k \text{ even}\\ \frac{1}{(k\pi)^2}, & \text{for } k \text{ odd}\end{cases}$$

Corollary 2.44. *Let $\mathcal{I}$ be an instance. Then*

$$\lim_{p\to\infty}|\widehat{f}(q)|^2=\frac{1}{\pi^2}$$

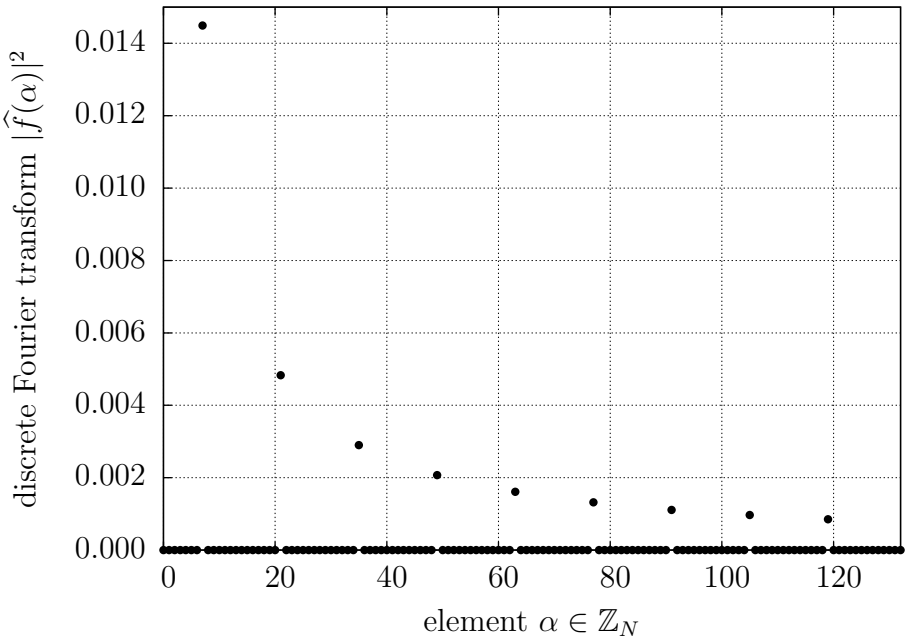


Figure 2: Values $|\widehat{f}(\alpha)|^2$ for $\alpha \in \mathbb{Z}_N$

In Figure 2 there are elements α from $\mathbb{Z}_N$ and the corresponding discrete Fourier transform $|\widehat{f}(\alpha)|^2$. The element with the biggest weight for the function f is q, while for f' is 0. Actually 0 for f' has weight even $\frac{1}{2}L_2(f')$. We want to find q, so we have decided to choose a function f instead of f' in SFT algorithm.

2.4 Algorithms for finding a significant coefficient

In this section we introduce two algorithms for finding a significant coefficient of function $f(x) = \mathrm{MSMB}(xq \bmod N) - 1/2$. Both of them assume that we have an access to an oracle O_f for function f, i.e. for arbitrary $x \in \mathbb{Z}$ we choose the oracle outputs the value $f(x)$. Thus the attack we will describe is a chosen plaintext attack. In practice O_f arises from the observations of a side channel in an RSA implementation as shown in Section 3. Notice that our goal is to find only one significant coefficient, namely q. Therefore we narrow down the searching interval from $[0, N]$ to $[2^{N_{bits}/2-1}, 2^{N_{bits}/2}]$ (N_{bits} is a number of bits of N) which contains q and no other significant coefficients of f (we show that this is an appropriate choice).

The first algorithm we present originates from Serge Vaudenay [12]. We improve the idea of this algorithm and some technical conditions to be faster than the original one.

Moreover, we bring a new algorithm for finding one significant coefficient. This algorithm (as well as the Vaudenay's one) is based on the properties of the discrete Fourier transform of the function f. It uses the results from the previous sections.

Convention:
Since in all applications of our algorithms presented in the thesis, all numbers p, q and N are large, we will make an assumption that $|\widehat{f}(x)|^2$ given by the instance $\mathcal{I}$ containing (p, q, N) is "almost equal" to $\lim_{p\to\infty} |\widehat{f}(x)|^2$. Indeed, for self-standing $|\widehat{f}(x)|^2$ the rounding error of this replacement is at most $\frac{1}{4p^2}$ (see Lemmas 2.36, 2.42). Besides that case we will use the assumption in the sum of form $\sum_{k=0}^{p-1} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2$ where the error is added p-times, so it cumulates for at most $p \cdot \frac{1}{4p^2}$ which is still "almost 0" for p being a 512-bit long number.

2.4.1 Searching interval

In our scenario, where $p, q \in \mathbb{N}$ are prime numbers, $N = pq$, N_{bits} is a number of bits of N, we want to narrow down the searching interval to contain q and no other significant coefficients of f. Appropriate choice of the interval is $[2^{N_{bits}/2-1}, 2^{N_{bits}/2}]$, as implemented in the OpenSSL [10].

2.4.2 Approximate GCD

In searching of q we will use Approximate GCD algorithm, in short AGCD by Howgrave-Graham [5]. He described how to find GCD of two numbers, from which first we know exactly and we have an estimate for the second one. Algorithm is based on lattices using LLL algorithm [6] by Lenstra–Lenstra–Lovász. We will use AGCD to find $\mathrm{GCD}(N, q')$ where q' is an estimate of q.

Lemma 2.45. *Let $N = pq$ and $q' = q + x$ where $|x| \leq \frac{N}{4}$. Then there exists polynomial time algorithm* AGCD *satisfying*

$$\mathrm{AGCD}(N, q') = \mathrm{GCD}(N, q)$$

Proof. See [5]. □

Observation 2.46. *In the limit case it is possible to find q by $\mathrm{AGCD}(N, q')$ even for $|x| = \frac{N}{4}$, but the lattice in* LLL *algorithm needs to be very large. On the other hand, with lattice size 5×5, the appropriate choice for bound of $|x|$ is $2^{\lceil 0.195 N_{bits} \rceil}$. For 1024 bit modulus N it means $|q - q'| < 2^{200}$ and for 2048 bit the bound is 2^{400}.*

2.4.3 Computing $\bar{\mathrm{W}}(c, N/L)$

Since f is a real function, oracle O_f is real, too. When we use only the real part of $\theta^{c(B_i-C_i)}$ (which is $\cos\left(\frac{2\pi}{N}\, c\,(B_i - C_i)\right)$) from definition

$$\bar{\mathrm{W}}(c, N/L) = \mathrm{Re}\left(\frac{1}{n}\sum_{i=1}^{n} f(A_i - B_i)\overline{f(A_i - C_i)}\,\theta^{c(B_i-C_i)}\right)$$

we can write

$$\bar{\mathrm{W}}(c, N/L) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{O}_f(A_i - B_i)\mathrm{O}_f(A_i - C_i)\cos\left(\frac{2\pi}{N}\, c\,(B_i - C_i)\right)$$

This simple observation speeds up the algorithm by factor 2.

Algorithm 1 Computing $\bar{\mathrm{W}}(c, N/L)$

Input: N, ε, γ, oracle O_f, interval $[a, b]$
Output: $\bar{\mathrm{W}}(c, N/L)$ satisfying Theorem 2.28 with probability 2ε

1: set $n = \frac{1}{2}\gamma^{-2}\frac{1}{\varepsilon}$, $L = \lceil \frac{N}{b-a+1} \rceil$ and $c = \lfloor \frac{a+b}{2} \rfloor$
2: pick $A_1, \ldots, A_n$ uniformly from $\mathbb{Z}_N$
3: pick $B_1, \ldots, B_n$ uniformly from $\mathbb{Z}_L$
4: pick $C_1, \ldots, C_n$ uniformly from $\mathbb{Z}_L$
5: return $\frac{1}{n}\sum_{i=1}^{n} \mathrm{O}_f(A_i - B_i)\mathrm{O}_f(A_i - C_i)\cos\left(\frac{2\pi}{N}\, c\,(B_i - C_i)\right)$

2.4.4 Estimating $|S_L(c - q)|^2$

If we restrict the searching interval to interval that contains only q and no other multiples of q, then in every level of the algoritm we can estimate the value $|S_L(c - q)|^2$ for given N, L and c using $\bar{\mathrm{W}}(c, N/L)$ (see the idea below). This information is useful for finding inteval(s) where q, our singificant coefficient, is located. The value $|S_L(c-q)|^2$ is equal to $|S_L(\alpha)|^2$ for some $\alpha \in [0, N/2]$ (and $-\alpha \in [-N/2, 0]$). If we focus on finding α (for example, by binary search), we can easily calculate $q = c - \alpha$.

The finding of the estimate of $|S_L(c - q)|^2$ is based on properties of the function $\widehat{f}$, especially

- $|\widehat{f}(0)|^2 = 0$ (Corollary 2.37)
- $|\widehat{f}(\beta)|^2 = 0$ for $\beta \neq kq$ (Corollary 2.39)

- $|\widehat{f}(kq)|^2 = 0$ for k even and
- $|\widehat{f}(kq)|^2 = \frac{1}{(k\pi)^2}$ for k odd (Lemma 2.43)

Lemma 2.47. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$ and $c \in [2^{N_{bits}/2-1}, 2^{N_{bits}/2}]$. Then*

$$\sum_{\substack{k=3 \\ k \text{ odd}}}^{p-2} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2 < \frac{\pi^2 - 8}{32} \cdot \frac{N}{L^2}$$

Proof.

$$\begin{aligned} \sum_{\substack{k=3 \\ k \text{ odd}}}^{p-2} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2 &\overset{2.43}{=} \sum_{\substack{k=3 \\ k \text{ odd}}}^{p-2} \frac{1}{(k\pi)^2} |S_L(c-kq)|^2 \\ &\overset{2.17(2)}{\leq} \frac{1}{\pi^2} \sum_{\substack{k=3 \\ k \text{ odd}}}^{p-2} \frac{1}{k^2} \cdot \frac{\pi^2}{16} \cdot \frac{N^2}{L^2} \cdot \frac{1}{|c-kq|_N^2} \end{aligned}$$

We want to estimate $|c-3q|_N^2$ from above, so we consider the smallest possible value: $c = 2^{N_{bits}/2}$ and $3q = 3.2^{N_{bits}/2-1} = 2^{N_{bits}/2} + 2^{N_{bits}/2-1}$. Now we have $|c-3q|_N^2 \geq (2^{N_{bits}/2-1})^2 \geq \frac{2^{N_{bits}}}{2^2} > \frac{N}{4}$. Thus

$$\sum_{\substack{k=3 \\ k \text{ odd}}}^{p-2} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2 < \frac{N^2}{16L^2} \sum_{\substack{k=3 \\ k \text{ odd}}}^{p-2} \frac{1}{k^2} \cdot \frac{4}{N} = \frac{N}{4L^2} \sum_{\substack{k=3 \\ k \text{ odd}}}^{p-2} \frac{1}{k^2}$$

The sum $\sum_{t=1}^{\infty} \frac{1}{t^2} = \frac{\pi^2}{6}$ is the well-known sum and for even elements, we compute

$$\sum_{t'=1}^{\infty} \frac{1}{(2t')^2} = \frac{1}{4} \sum_{t'=1}^{\infty} \frac{1}{t'^2} = \frac{1}{4} \cdot \frac{\pi^2}{6}$$

so for odd elements, we get

$$\sum_{\substack{t'=1 \\ t' \text{ odd}}}^{\infty} \frac{1}{t'^2} = \frac{3}{4} \cdot \frac{\pi^2}{6} = \frac{\pi^2}{8}$$

Notice that for our purpose we can exclude the member for $t' = 1$

$$\sum_{\substack{k=3 \\ k \text{ odd}}}^{p-2} \frac{1}{k^2} < \sum_{\substack{k=3 \\ k \text{ odd}}}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{8} - 1 = \frac{\pi^2 - 8}{8}$$

Finally, we have

$$\sum_{\substack{k=3\\ k\,\mathrm{odd}}}^{p-2} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2 < \frac{N}{4L^2} \cdot \frac{\pi^2-8}{8} = \frac{\pi^2-8}{32} \cdot \frac{N}{L^2}$$

□

Theorem 2.48. *Let $\mathcal{I}$ be an instance, $L \in \mathbb{Z}_N \setminus \{0\}$, $c \in [2^{N_{bits}/2-1}, 2^{N_{bits}/2}]$ and $\Omega = \frac{\pi^2-8}{32} \cdot \frac{N}{L^2}$. Further, let ε, γ are chosen parameters as in Theorem 2.28 and $\bar{\mathrm{W}}(c, N/L)$ is computed from i.i.d. variables. Then the following holds with probability $1-2\varepsilon$*

$$\frac{\bar{\mathrm{W}}(c,N/L) - \gamma - \Omega}{|\widehat{f}(q)|^2} \leq |S_L(c-q)|^2 \leq \frac{\bar{\mathrm{W}}(c,N/L) + \gamma}{|\widehat{f}(q)|^2}$$

Proof. We use results from the section above, especially Corollary 2.37 and Corollary 2.39.

$$\begin{aligned} \bar{\mathrm{w}}(c,N/L) &= \sum_{\alpha \in \mathbb{Z}_N} |\widehat{f}(\alpha)|^2 |S_L(c-\alpha)|^2 = \sum_{\substack{k=1\\ k\,\mathrm{odd}}}^{p-2} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2 \\ &= |\widehat{f}(q)|^2 |S_L(c-q)|^2 + \sum_{\substack{k=3\\ k\,\mathrm{odd}}}^{p-2} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2 \end{aligned}$$

Now from Lemma 2.47, we have

$$\sum_{\substack{k=3\\ k\,\mathrm{odd}}}^{p-2} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2 \leq \Omega$$

and since $|\bar{\mathrm{W}}(c,N/L) - \bar{\mathrm{w}}(c,N/L)| \leq \gamma$, i.e.

$$\bar{\mathrm{W}}(c,N/L) - \gamma \leq \bar{\mathrm{w}}(c,N/L) \leq \bar{\mathrm{W}}(c,N/L) + \gamma$$

with probability 2ε (Theorem 2.28), we get

$$\bar{\mathrm{W}}(c,N/L) - \gamma \leq |\widehat{f}(q)|^2 |S_L(c-q)|^2 + \sum_{\substack{k=3\\ k\,\mathrm{odd}}}^{p-2} |\widehat{f}(kq)|^2 |S_L(c-kq)|^2 \leq \bar{\mathrm{W}}(c,N/L) + \gamma$$

$$\bar{\mathrm{W}}(c,N/L) - \gamma - \Omega \leq |\widehat{f}(q)|^2 |S_L(c-q)|^2 \leq \bar{\mathrm{W}}(c,N/L) + \gamma$$

$$\frac{\bar{\mathrm{W}}(c,N/L) - \gamma - \Omega}{|\widehat{f}(q)|^2} \leq |S_L(c-q)|^2 \leq \frac{\bar{\mathrm{W}}(c,N/L) + \gamma}{|\widehat{f}(q)|^2}$$

□

Corollary 2.49. *For an instance $\mathcal{I}$, $L \in \mathbb{Z}_N \setminus \{0\}$, $c \in [2^{N_{bits}/2-1}, 2^{N_{bits}/2}]$ and $\gamma \in \mathbb{R}$, we get*

$$\left(\bar{\mathrm{W}}(c, N/L) - \gamma - \frac{\pi^2 - 8}{32} \cdot \frac{N}{L^2}\right) \pi^2 \leq |S_L(c-q)|^2 \leq \left(\bar{\mathrm{W}}(c, N/L) + \gamma\right) \pi^2$$

Proof. Follows from Lemma 2.47 and Corollary 2.44 ($|\widehat{f}(q)|^2 = \frac{1}{\pi^2}$). □

Remark 2.50. Notice that for calculating the lower and the upper bounds of $|S_L(c-q)|^2$ (Corollary 2.49) we do not need to know prime numbers p, q, we only need to know N, L, c, ε and γ. We will use this knowledge because it is very useful for our attack.

2.4.5 Intervals containing q

As we showed above, from computing $\bar{\mathrm{W}}(c, N/L)$ we can estimate $|S_L(c-q)|^2$. Now, we can continue even further. From bounds on $|S_L(c-q)|^2$ we compute bounds on $|c-q|$ (for example by binary search) and since we know the value c, we can determine interval(s) in which q lies.

Algorithm 2 Interval(s) containing q *intq()*

Input: N, ε, γ, $L \in \mathbb{Z}_N$, $c \in \mathbb{Z}_N$, $\bar{\mathrm{W}} = \bar{\mathrm{W}}(c, N/L)$ for interval $[a, b]$
Output: interval(s) where q lies

1: $\Omega = \frac{\pi^2-8}{32} \cdot \frac{N}{L^2}$
2: $B_l = (\bar{\mathrm{W}} - \gamma - \Omega)\,\pi^2$, $B^u = (\bar{\mathrm{W}} + \gamma)\,\pi^2$ // $B_l \leq |S_L(c-q)|^2 \leq B^u$
3: $\xi = \mathrm{SL2inv}(B_l)$, $\eta = \mathrm{SL2inv}(B^u)$
4: **if** $\eta = 0$ **then**
5: int $= \{[c-\xi, c+\xi]\}$
6: **else**
7: int $= \{[c-\xi, c-\eta], [c+\eta, c+\xi]\}$
8: **end if**
9: **return** int

In Algorithm 2, SL2inv is a function which finds $|x|$ for value $|S_L(x)|^2$. When we start searching for q in interval $[a, b]$, this algorithm outputs new (smaller) interval(s) where q lies. If B^u, the upper bound on $|S_L(c-q)|^2$, is equal or greater than 1, the new interval is only one and it is situated around c, see Figure 3.

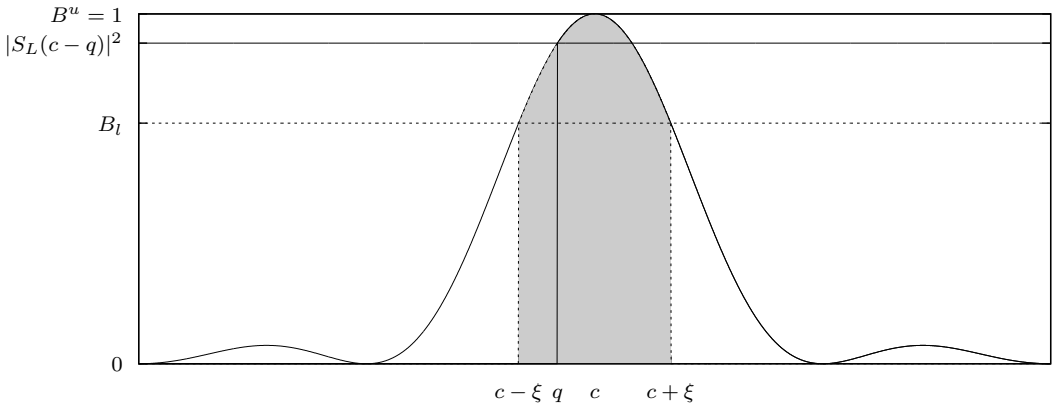


Figure 3: q-containing interval if the upper bound $B^u \geq 1$

On the other hand, if the upper bound is less than 1, new intervals are two and of course, q is situated only in one of them, see Figure 4.

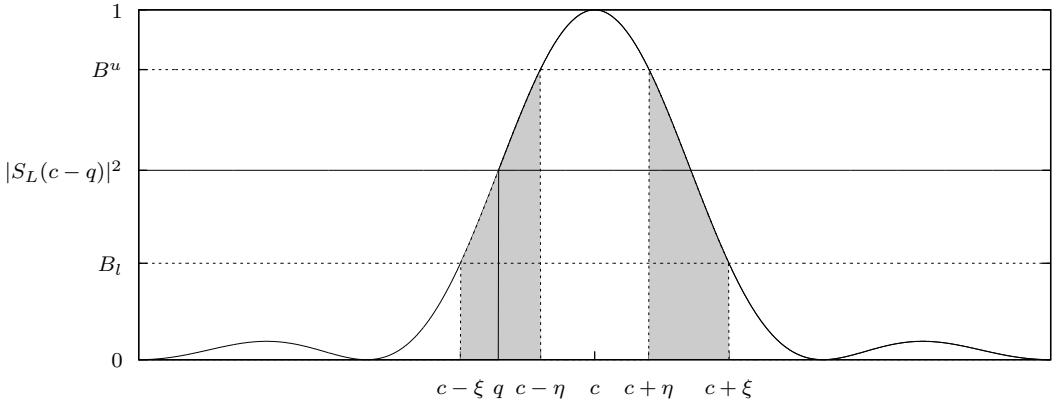


Figure 4: q-containing intervals if the upper bound $B^u < 1$

2.4.6 Improved Vaudenay's algorithm

The original Vaudenay's algorithm is recursive. He proposed to enlarge the searching interval by one third of its length (from δ to $\frac{4}{3}\delta$) and split it into three overlapping intervals of size $\frac{2}{3}\delta$ centered at points $-\frac{\delta}{3}, 0$ and $\frac{\delta}{3}$ relative to the center of the original interval. Then in each subinterval he calculated $\bar{\mathrm{W}}(c, N/L)$ where c is the center point of the subinterval and finally picked that one with the biggest $\bar{\mathrm{W}}$ to continue recursively on this subinterval. Since Vaudenay's algorithm has not been published yet, we have decided not to describe it in more details.

We bring a new insight to this algorithm. Instead of picking one interval from three subintervals based on the maximum value of $\bar{\mathrm{W}}$, we calculate an estimate of $|S_L(c-q)|^2$ for each of this three subintervals and find intervals,

where q, our significant coefficient, may lie (Algorithm 2). Finally, we intersect these potentionally q-containing intervals and continue bt searching only on the intersection. At the end the searching interval is so small that we can easily find q by Howgrave-Graham's AGCD algorithm [5].

As we will show in Section 4, the new approach of Vaudenay's algorithm is about five times faster than the original one.

Algorithm 3 Improved Vadenay's algorithm for finding a single SFT

Input: N, ε, γ, oracle O_f, interval $[a, b]$ where $a = 2^{\frac{N_{bits}}{2}-1}$, $b = 2^{\frac{N_{bits}}{2}}$
Output: singnificant coefficient q

$n = \frac{1}{2}\gamma^{-2}\frac{1}{\varepsilon}, \quad l = 0.195 N_{bits}$
while $|b - a| > 2^l$ **do**
  $\delta = b - a + 1, \quad L = \lceil \frac{N}{\delta} \rceil$
  for $i = 1$ to 3 **do**
    $a_i = a + \lfloor \frac{\delta}{6}(2i - 3) \rfloor, \quad b_i = a + \lceil \frac{\delta}{6}(2i + 1) \rceil - 1$
    $c_i = \lfloor \frac{a_i + b_i}{2} \rfloor$
    $\bar{\mathrm{W}} = \bar{\mathrm{W}}(c_i, N/L)$ // Alg. 1 for interval $[a_i, b_i]$
    $\mathrm{int}_i = intq(\bar{\mathrm{W}})$ // Alg. 2 for $\bar{\mathrm{W}}(c_i, N/L)$
  end for
  $[a, b] = \bigcap_{i=1}^{3} \mathrm{int}_i$
end while
return AGCD(N, a)

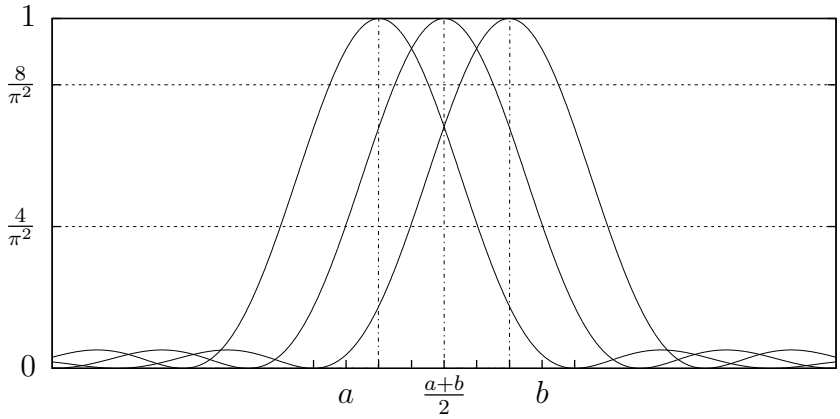


Figure 5: In one iteration of improved Vaudenay's algorithm three values $\bar{\mathrm{W}}$ and consequently three pairs of bounds on $|S_L(c - q)|^2$ are computed

2.4.7 New algorithm

Now, we will describe a new algorithm for finding a significant coefficient of function f. It is recursive, as well. In the first iteration we start with searching interval $[a,b] = [2^{N_{bits}/2-1}, 2^{N_{bits}/2}]$ and settings $c = \lfloor \frac{a+b}{2} \rfloor$, $L = \lceil \frac{N}{b-a+1} \rceil$. In each step we compute one $\bar{\mathrm{W}}(c, N/L)$ on every input interval. Using Algorithm 2 we get potentionally q-containing subinterval(s). Finally, we intersect these subintervals with all intervals we have got so far. Input intervals for the next step are all intervals in this intersection. As in previous algorithm at the end we use AGCD.

We use in new algorithm:

- *intIn, intOut* are arrays of intervals
- function *len* means the length of array, i.e. count of intervals in intIn
- function *intersection* finds interval(s) - the intersection of two interval arrays, i.e. x is in this intersection if and only if x is at least one interval from the first array and at least one interval from second array
- function $index\bar{\mathrm{W}}_{\max}$ finds such index j for which $\bar{\mathrm{W}}_j$ is the biggest

Algorithm 4 New algorithm for finding a single SFT

Input: N, ε, γ, oracle O_f, interval $[a,b]$ where $a = 2^{\frac{N_{bits}}{2}-1}$, $b = 2^{\frac{N_{bits}}{2}}$
Output: singnificant coefficient q

$n = \frac{1}{2}\gamma^{-2}\frac{1}{\varepsilon}, \quad l = 0.195 N_{bits}$
intIn $= \{[a,b]\}$
repeat
  intOut $= \{\}$
  for $i = 0$ to len(intIn) $- 1$ **do**
    $[a,b] =$ intIn$[i]$
    $c_i = \lfloor \frac{a+b}{2} \rfloor, \quad L = \lceil \frac{N}{b-a+1} \rceil$
    $\bar{\mathrm{W}}_i = \bar{\mathrm{W}}(c_i, N/L)$ // Alg. 1 for interval $[a,b]$
    intOut $=$ intOut $\cup\, intq(\bar{\mathrm{W}}_i)$ // Alg. 2 for $\bar{\mathrm{W}}(c_i, N/L)$
  end for
  intIn $=$ intersection(intIn, intOut)
until for each $[a,b] \in$ intIn: $b - a < 2^l$
$j = \mathrm{index}\bar{\mathrm{W}}_{\max}(\bar{\mathrm{W}}_0, \ldots, \bar{\mathrm{W}}_i)$ // $\bar{\mathrm{W}}_j = \max_{0 \le k \le i} \bar{\mathrm{W}}_k$
return AGCD(N, c_j)

2.4.8 Technical improvements

There are a few improvements that can increase efficiency of our algorithms. Their description follows.

Improvement 1
In both algorithms we compute bounds on $|S_L(c-q)|^2$ for interval $[a, b]$ where c is the centre. Notice that if $q \in [a, b]$ then $|S_L(c-q)|^2$ has to be greater than $\frac{4}{\pi^2}$ since points with the smallest value of $|S_L(c-x)|^2$ are $a = c - \frac{N}{2L}$ and $b = c + \frac{N}{2L}$ and it holds $|S_L(c-a)|^2 = |S_L(c-b)|^2 \geq \frac{4}{\pi^2}$ (Lemma 2.19 (2)). Thus we can take this into account in our algorithms, i.e. when the upper bound on $|S_L(c-q)|^2$ is smaller than $\frac{4}{\pi^2}$, we omit the interval $[a, b]$ in the next steps of algorithms.

Now, we focus on improved Vaudenay's algorithm. We can afford to decrease the number n of summands (i.e. side channel observations) in computation of $\bar{\mathrm{W}}(c, N/L)$ (see Algorithm 1), so that the algorithm will be faster. This may cause the estimates on $|S_L(c-q)|^2$ to be "bad". To handle problems with bad estimates we propose to add two new heuristic features described in Improvements 2 and 3.

Improvement 2 (Heuristic)
A bad estimate can cause the intersection in step 10 in Algorithm 3 to be empty. When this happens we decide to repeat the same iteration again.

Improvement 3 (Heuristic)
Due to a bad estimate, the computation can get into an impasse, i.e. the intersection in step 10 in Algorithm 3 may not contain q, so consequently in the next iteration, q is not in the input interval $[a, b]$. In this iteration all bounds on $|S_L(c-q)|^2$ will be smaller than $\frac{4}{\pi^2}$ for the same reason as described in Improvement 1. After recognizing that, we return one iteration back.

3 RSA-CRT algorithm and the Side Channel

RSA is an algorithm for public-key cryptography published in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman. The current version [9] is available in PKCS #1 standard v2.1 . It is based on difficulty to factorize a big number to the product of its divisors.

In this section we present an implementation of RSA with Montgomery multiplication using Chinese Remainder Theorem (CRT). We also show the side channel which can be revealed from the specific instance of RSA, originally described by Hlaváč. In his article [3], he presented known plaintext only attack with the side channel using lattices, especially solving Hidden Number Problem (HNP) by LLL algorithm [6]. On the other hand, motivated by Akavia [1], we describe adaptive chosen plaintext attack by solving HNP with finding SFT coefficient.

We will use Tomoeda's notation while describing RSA-CRT and Montgomery operations.

3.1 RSA-CRT with Montgomery Multiplication

Definition 3.1. Let $p, q \in \mathbb{N}$ be two different randomly chosen prime numbers, modulus $N = pq$ and Euler funtion $\phi(N) = (p-1)(q-1)$. Further let $e \in \mathbb{N}$ satisfy $e \in (1, \phi(N))$ and $\mathrm{GCD}(e, \phi(N)) = 1$ and $d \in \mathbb{N}$ from interval $(1, \phi(N))$ with the following condition $d = e^{-1} \bmod \phi(N)$. Then, we define

- (N, e) as RSA public key and
- (N, d) as RSA private key

Prime numbers p and q are chosen by an appropriate way, for example, difference $|p - q|$ should be large enough, numbers $p - 1$ and $q - 1$ should have at least one big factor, etc. At present, modulus N of length at least 1024 bits is used.

3.1.1 RSA signing using CRT

For a given message m, the RSA signature is computed by the operation

$$s = m^d \bmod N$$

In RSA using CRT we first define $d_p = d \bmod (p-1)$, $d_q = d \bmod (q-1)$, $m_p = m \bmod (p-1)$, $m_q = m \bmod (q-1)$ and $p_{inv} = p^{-1} \bmod q$ and the RSA signing is as follows

$$\begin{aligned} s_p &= (m_p)^{d_p} \bmod p \\ s_q &= (m_p)^{d_q} \bmod q \\ s &= ((s_q - s_p)p_{inv} \bmod q)p + s_p \end{aligned}$$

3.1.2 Montgomery Exponentiation

Montgomery exponentiation is used in calculating s_p and s_q for its quickness. First, numbers are converted to the Montgomery representation and instead of very expensive integer division there is a division by Montgomery constant $R = 2^{\lceil \frac{N_{bits}}{2} \rceil}$. It is very fast in the binary representation. Definition of Montgomery representation of integers and algorithms of multiplication and exponentiation follow.

Definition 3.2. For instance $\mathcal{I}$ and $R = 2^{\lceil \frac{N_{bits}}{2} \rceil}$ we define Montgomery representation $\mu(x)$ of $x \in \mathbb{Z}_p$ as follows

$$\mu(x) = xR \bmod p$$

Lemma 3.3. *Let $\mathcal{I}$ be an instance and $x, y \in \mathbb{Z}_p$. Then $\mu(xq) = \mu(x) * \mu(y)$, where $a * b = abR^{-1} \bmod p$ is Montgomery multiplication.*

Proof.

$$\begin{aligned} \mu(x) * \mu(y) &= (xR) \bmod p * (yR) \bmod p = (xR)(yR)R^{-1} \bmod p \\ &= xyR \bmod p = \mu(xy) \end{aligned}$$

□

Algorithm 6 is a binary exponentiation with help of Montgomery representation of integers. Dummy operation in step 8 is a prevention of Single Power Analysis (SPA) attack. Correctness of the Algorithm 5 is described in [7]. The sixth step is called Final Subtraction (FS). We assume it is possible to obtain the amount of FS during one operation *mont(z,z,p)* and we consider this information as the side channel for our attack.

Algorithm 5 Montgomery multiplication *mont()*

Input: $x, y \in \mathbb{Z}_p$
Output: $v = xyR^{-1} \bmod p$ // x, y, v are in Montgomery representation

1: $s = xy$
2: $t = s(-p^{-1}) \bmod R$
3: $g = s + tp$
4: $v = g/R$
5: **if** $v > p$ **then**
6: $\quad v = v - p$ // FS - final subtraction
7: **end if**
8: **return** v

Algorithm 6 Montgomery exponentation *expmont()*

Input: $m, p, d = (d_{n-1}d_{n-2}\ldots d_1 d_0)_2$
Output: $x = m^d \bmod p$

1: $u = mR \bmod p$
2: $z = u$
3: **for** $i = n - 2$ to 0 **do**
4: $\quad z = mont(z, z, p)$
5: $\quad$ **if** $d_i == 1$ **then**
6: $\quad\quad z = mont(z, u, p)$
7: $\quad$ **else**
8: $\quad\quad z' = mont(z, u, p)$ // dummy multiplication
9: $\quad$ **end if**
10: **end for**
11: $z = mont(z, 1, p)$
12: **return** z

3.2 The Side Channel

In this part we describe the side channel. It leads to the function $f(x) = \text{MSMB}(xq \bmod N)$ and it is based on Tomoeda's estimate [11] for the number n_i of FS and consequently on Hlaváč's conversion described in [3].

3.2.1 Tomoeda's estimate

Observation 3.4. *For $p, q \in \mathbb{N}$ prime numbers, $N = pq$, message $m \in \mathbb{Z}_N$ and k observations, consider n_i as #FS in operation $(m_{p,i})_p^d \bmod p$, where $m_{p,i} = m_i \bmod N$ and $i \in [1, k]$. Further, let $n_{max} = \max_{1 \leq i \leq k} n_i$, $n_{min} = \min_{1 \leq i \leq k} n_i$ and R be a Montgomery constant. Then we can approximate*

$$\frac{m_i R \bmod p}{p} \approx \frac{n_i - n_{min}}{n_{max} - n_{min}}$$

3.2.2 Hlaváč's conversion

Lemma 3.5. *For parameters from the previous lemma and substitutions $t_i = m_i R \bmod N$ and $u_i = \frac{n_i - n_{min}}{n_{max} - n_{min}} N$, we obtain*

$$t_i q \approx u_i \pmod N$$

Proof. There exists $k_i \in \mathbb{Z}$ such that $m_i R \bmod p = m_i R - k_i p$. Then we multiply the whole approximation by N

$$m_i R q - k_i N \approx \frac{n_i - n_{max}}{n_{max} - n_{min}} N$$

When we substitute $t_i = m_i R \bmod N$ and $u_i = \frac{n_i - n_{min}}{n_{max} - n_{min}} N$, we get

$$\begin{aligned} t_i q + k_i' N &\approx u_i \\ t_i q &\approx u_i \pmod N \end{aligned}$$

□

3.2.3 Our side channel

If we are able to obtain the total number of FS during the exponentiation $m_p^{d_p} \bmod p$, then since $t_i q \approx u_i \pmod N$, we can choose a function MSMB for which it holds

$$\mathrm{MSMB}(t_i q \bmod N) \approx \mathrm{MSMB}(u_i \bmod N)$$

We define a function $f'(x) = \mathrm{MSMB}(xq \bmod N)$ and set our oracle as follows. When someone "asks" for the value $f'(t_i) = \mathrm{MSMB}(t_i q \bmod N)$ the oracle will answer the known value $\mathrm{MSMB}(u_i \bmod N)$ instead of the uknown value $\mathrm{MSMB}(t_i q \bmod N)$. In practice they are equal with probability 95.8% (see 4.1) for 1024 bit long RSA modulus. The choice of MSMB function is very convenient for the SFT algorithm. Let us remind that we use function $f(x) = \mathrm{MSMB}(xq \bmod N) - 1/2$ instead of $f'(x)$.

In the Algorithm 7 we show how the oracle O_f answers the value $\mathrm{MSMB}(u_i \bmod N) - 1/2$ for some t_i. During the computation of u_i we need to know values n_{min} and n_{max}, the minimum and the maximum numbers of FS during k observations. In practice $n_{min} = 0$ (see [3]) and as Primas showed in [8] it is better to use $2 \cdot n_{avg}$ where $n_{avg} = \frac{1}{k}\sum_{i=1}^{k} n_i$ instead of n_{max}. First, we let sign a few messages to get the average number of FS. In the Algorithm 7 the function $montSign(m_i)$ outputs the number of FS during RSA signing, especially during Montgomery exponentiation $m_{p,i}^{d_p} \bmod p$.

Algorithm 7 Oracle O_f $\quad \mathrm{O}_f(t_i)$

Input: $N, R = 2^{\lceil \frac{N_{bits}}{2} \rceil}, t_i \in \mathbb{Z}_N$
Output: $\mathrm{MSMB}(t_i q \bmod N) - 1/2$ with probability 95.8%

1: $m_i = t_i R^{-1} \bmod N$
2: $n_i = montSign(m_i)$ // $n_i = \#\mathrm{FS}$
3: $u_i = \frac{n_i}{2n_{avg}} N$
4: **return** $\mathrm{MSMB}(u_i \bmod N) - 1/2$

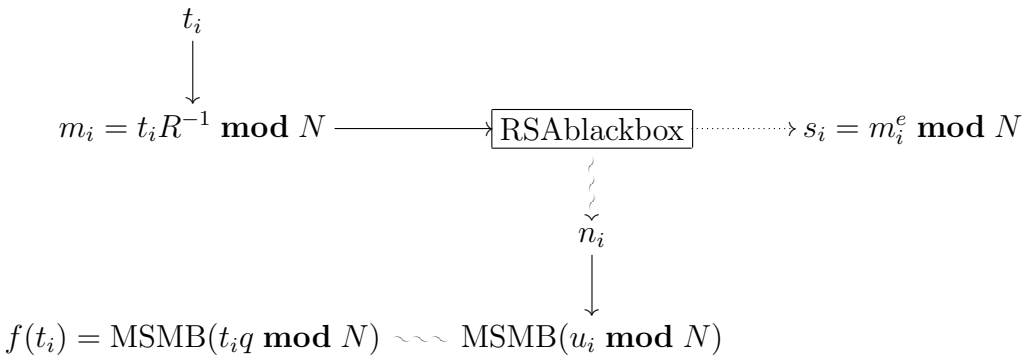


Figure 6: Oracle answer to the query t_i

3.2.4 Algorithms summary

For better understanding we will summarize all steps in both algorithms. Suppose we have a blackbox which for message m outputs its RSA signature $s = m^e \bmod N$ for an unknown RSA instance. What we know about this blackbox is that it uses Montgomery exponentiation for RSA signing and for each message m_i reveals n_i, the number of FS during exponentiation $\bmod p$.

For functionality of our oracle O_f described in Algorithm 7 we need to know n_{avg}, the average number of FS during k observations, so first we let our blackbox sign k random messages (we apply $k = 1000$) and from $n_1, \ldots, n_k$ compute n_{avg} as their mean. Now, for computation $\bar{\mathrm{W}}$, we can use oracle O_f.

The second step is to run an algorithm for finding a single SFT coefficient, improved Vaudenay's algorithm (see Alg. 3) or a new algorithm (see Alg. 4) with our oracle O_f. Besides the RSA public modulus N other input parameters are ε and γ which come from Theorem 2.28. They have impact on the precision we want in computation of $\bar{\mathrm{W}}$ and how many observations are needed. The choice of these parameters we be discussed in subsection 4.4. During the SFT algorithm we can apply some technical improvements descibed in subsection 2.4.8, especially Improvement 1 for both algorithms to avoid cases where the current searching interval does not contain q, our significant coefficient, or Improvements 2 and 3 for improved Vaudenay's algorithm to be faster.

We will stop the SFT algorithm when the searching interval(s) is(are) so small that we can apply AGCD algorithm by Howgrave-Graham (see subsection 2.4.2) to find q.

4 Practical results

After the theoretical part we present practical results. We look at probability that our oracle works as we have expected and how long the computation of $\bar{\mathrm{W}}$ takes for different lengths of n. We discuss the choice of input parameters to the SFT algorithms. We also compare the relative amounts of time which the algorithm spends by generating the side channel information and the rest of the algorithm. We show the difference between the original Vaudenay's algorithm and its improvement that we have proposed. Finally, we present time results of both algorithms.

For our experiments we used computer with Intel Core 2 Duo 3.00GHz. The following results are for 1024 bit modulus N.

4.1 Probability which the oracle O_f works with

First we present the probability of functionality of our oracle O_f described in Algorithm 7. For 10 instances of RSA we let sign 1000 random messages and compute how many times the equation below holds

$$\mathrm{MSMB}(t_i q \textbf{ mod } N) = \mathrm{MSMB}(u_i \textbf{ mod } N)$$

RSA instance	**success rate**
1	96.8%
2	95.9%
3	96.0%
4	96.9%
5	96.5%
6	94.7%
7	94.9%
8	95.7%
9	95.1%
10	95.5%
mean	**95.8%**

Table 1: Functionality of oracle O_f

4.2 Duration of computing $\bar{\mathrm{W}}$ for different n

In the graph below we show how long the computation of $\bar{\mathrm{W}}$ takes for different lengths of n. These results also affect the choice of input parameters to the SFT algorithm. We see that the dependence is linear.

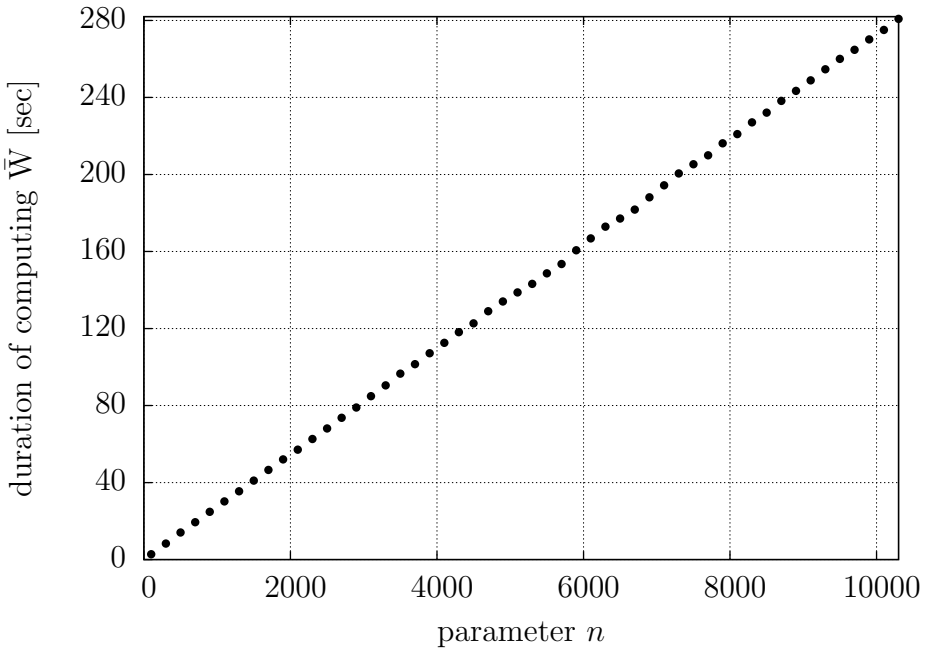


Figure 7: Duration of computing $\bar{\mathrm{W}}$ for parameter n

4.3 The side channel generation vs. the SFT algorithm

The generation of the side channel information is very time consuming operation, so that we are interested in finding out its proportion in the whole algorithm running time. Of course, it depends on parameter n. For $n = 300$ it starts on 93% and it rising to 99.5% for $n = 9300$.

4.4 The choice of input parameters for SFT algorithm

There are two input parameters for the SFT algorithm, ε and γ. They affect n, the number of values that are added in computing $\bar{\mathrm{W}}$, definition of which is $n = \frac{1}{2}\gamma^{-2} \ln \frac{1}{\varepsilon}$. Now, we remind where they occur and what they mean.

First of all, they come from Theorem 2.28. With little modification we have

$$\Pr[|\,\bar{\mathrm{W}}(c, N/L) - \bar{\mathrm{w}}(c, N/L)| < \gamma] > 1 - 2\varepsilon$$

Thus with probability $(1-2\varepsilon)$, values $\bar{\mathrm{W}}$ and $\bar{\mathrm{w}}$ differ by less than γ. We can set γ and ε very small, but then n is very large and as we have shown in Table 2, it has a big impact on the amount of side channel measurements and also on the duration of the algorithm.

On the other hand, we use γ in Algorithm 2 for estimate of $|S_L(c-q)|^2$, especially for its lower and upper bounds

$$B_l = (\bar{\mathrm{W}} - \gamma - \Omega)\,\pi^2, \quad B^u = (\bar{\mathrm{W}} + \gamma)\,\pi^2$$

Because of the fact that the oracle O_f does not give correct values in 100% cases and also $\bar{\mathrm{W}}$ is calculated with probability $(1-2\varepsilon)$, the value γ does not have to be too small to cover these imprecisions. On the contrary, it also may not be too large because in that case we get a big range for estimate of $|S_L(c-q)|^2$.

Finally, we have decided to set ε to 0.01, so the probability of correct computation of $\bar{\mathrm{W}}$ is 98%. The reason is that even one bad estimate of $|S_L(c-q)|^2$ can destroy the SFT algorithm.

For fixed $\varepsilon = 0.01$, we consider acceptable γ from 0.01 to 0.02 (it sets the range from 0.197 to 0.394 for estimate of $|S_L(c-q)|^2$ when γ is multiplied by π^2). See γ and the corresponding n in Table 2 below.

γ	n
0.010	23026
0.011	19030
0.012	15991
0.013	13625
0.014	11748
0.015	10234
0.016	8995
0.017	7968
0.018	7107
0.019	6379
0.020	5757

Table 2: Choosing parameters: γ and the corresponding n for fixed $\varepsilon = 0.01$

Other question is how fast the searching interval decreases during the algorithm. Because of the different characters of mentioned algorithms and the fact that computation of $\bar{\mathrm{W}}$'s takes at least 95% of the running time, we have decided to define unit "bits/$\bar{\mathrm{W}}$" which represents number of bits that are revealed during computation of one $\bar{\mathrm{W}}$. It is only analytical unit because

very often it is neccessary to compute more $\bar{W}$'s for one level of algorithm. In the Table 3 we present bits/$\bar{W}$ for different γ if our side channel is correct in 100% cases.

Finally, we decided for the trade-off and set $\gamma = 0.0155$ yielding $n = 9585$.

γ	bits/$\bar{W}$	
	impr. Vaudenay's alg.	**new SFT algorithm**
0.010	1.21	1.35
0.011	1.19	1.28
0.012	1.14	1.21
0.013	1.10	1.17
0.014	1.07	1.11
0.015	1.03	1.05
0.016	1.00	1.00
0.017	0.97	0.95
0.018	0.94	0.92
0.019	0.92	0.86
0.020	0.89	0.82

Table 3: Values γ and the corresponding bits/$\bar{W}$ for both algorithms

4.5 Original vs. improved Vaudenay's algorithm

Now we show the difference between the original Vaudenay's algorithm and the improved one. For fixed ε, γ and consequently n we present the results in unit bits/$\bar{W}$ for 5 instances of RSA. It is seen that improved version of Vaudenay's algorithm for our special function f is about five times faster than the original one. We also add the results for the new SFT algorithm which is even a bit faster than the improved Vaudenay's.

RSA ins.	bits/$\bar{W}$		
	orig. Vaudenay's	**impr. Vaudenay's**	**new SFT**
1	0.196	1.019	1.067
2	0.197	1.013	1.064
3	0.193	1.000	1.071
4	0.195	1.006	1.060
5	0.195	1.006	1.046
mean	**0.195**	**1.009**	**1.062**

Table 4: Comparison of speed of decreasing the searching interval

4.6 Timing results for improved Vaudenay's algorithm

In the next two parts of the work we present the timing results of both algorithms. The Figure 8 is for improved Vaudenay's algorithms, further the Figure 9 stands for the case when we have used heuristic improvements described in 2.4.8 and the last one (Figure 10) represents timing results for our new algorithm. From these results we see that the heuristic version of improved Vaudenay's algorithm is the fastest one, the whole attack takes only about an hour. On the other hand, we notice that there is almost no time difference between our new algorithm and the improved Vaudenay's one. All measurements are for 1024 bit RSA.

4.6.1 Improved Vaudenay's algorithm

First, we present improved Vaudenay's algorithm. During the computation we used Improvement 1 from 2.4.8 and the Howgrave-Graham's AGCD algorithm at the end. Timing results are shown in Figure 8.

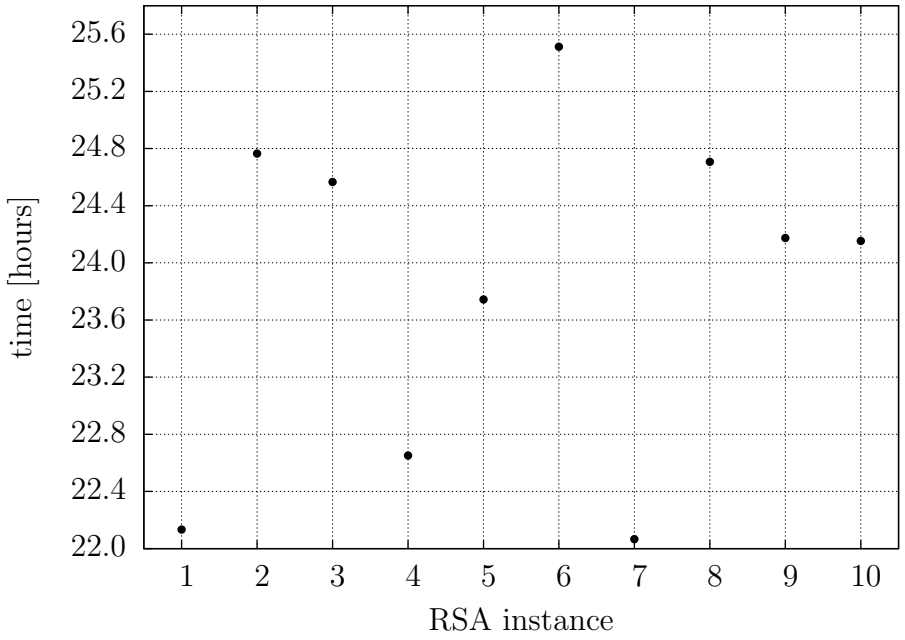


Figure 8: Timing results of improved Vaudenay's algorithm for 1024-bit RSA

4.6.2 Heuristically improved Vaudenay's algorithm

We used Improvement 1, heuristic Improvements 2 and 3 from subsection 2.4.8 and also the AGCD algorithm. As the best option of decreased parameter n, the choice $n/30$ appears, so the total number of requested observations is reduced to $1/30$. Since the parameter n is quite small, the whole algorithm is really fast. Timing results are available in Figure 9.

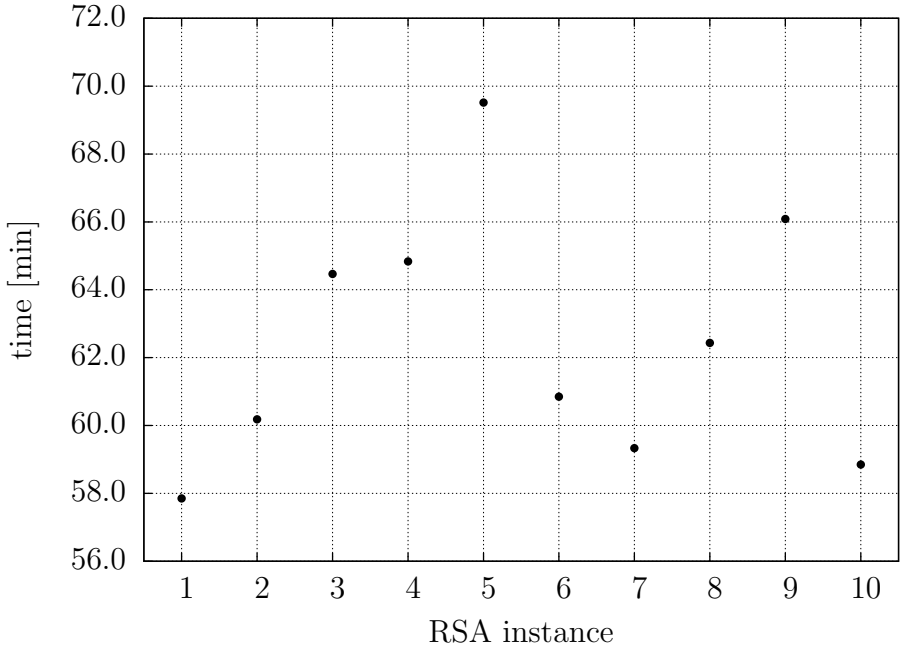


Figure 9: Timing results of heuristic Vaudenay's algorithm for 1024-bit RSA

4.7 Timing results for the new algorithm

Finally, we present results for our new algorithm. At the end the AGCD algorithm is also applied. Figure 10 shows the timing results.

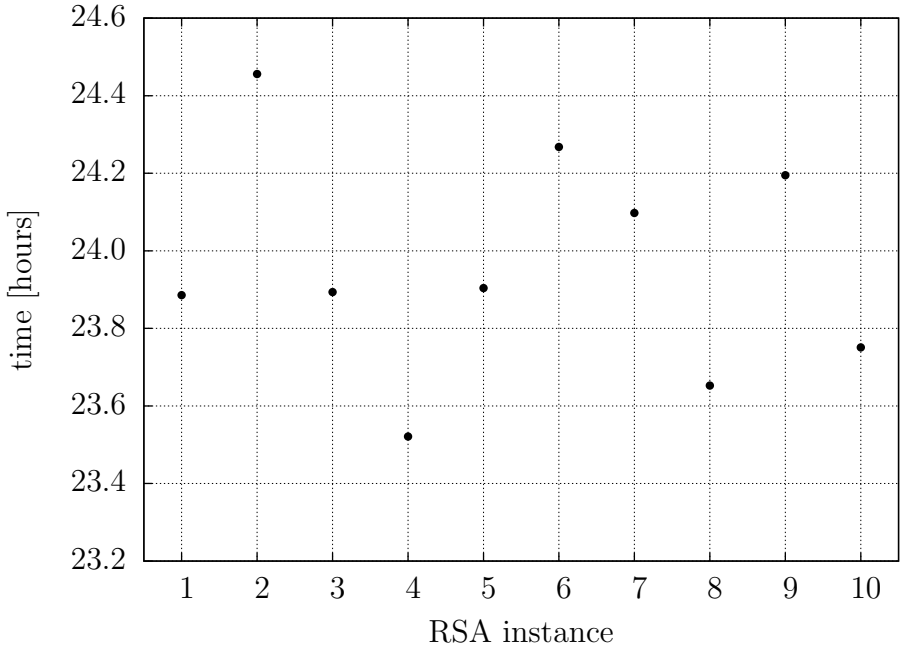


Figure 10: Timing results of new algorithm for 1024-bit RSA

The timing results of our new algorithm belong to the interval 23.5 - 24.5 with the mean 24 hours. There is no big difference between the mean of the new algorithm and the mean of improved Vaudenay's algorithm.

5 Conclusion

We have presented an improved Vaudenay's and a new algorithm for finding a significant Fourier transform coefficient for one special function and described its theoretical basis. We have shown how they can be applied for an adaptive chosen plaintext attack on RSA-CRT signing using Montgomery multiplication. We have also presented the practical results which imply that improved Vaudenay's algorithm is five times faster than the original one. By several technical improvements for 1024 bit modulus, we have achieved the running time of the algorithm to be only about an hour.

Bibliography

[1] Akavia A.: Solving Hidden Number Problem with One Bit Oracle and Advice. *Advances in Cryptology – Proceedings of CRYPTO '09*, Springer-Verlag, 2009, 337–354.

[2] Čížek V.: *Diskrétní Fourierova Transformace a její použití*. Praha, 1981.

[3] Hlaváč M.: Known–Plaintext–Only Attack on RSA–CRT with Montgomery Multiplication. *Cryptographic Hardware and Embedded Systems – Proceedings of CHES 2009*, Lecture Notes in Computer Science 5747, Springer-Verlag, 2009, 128–140.

[4] Hoeffding W.: Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963, 13–30.

[5] Howgrave-Graham N.: Approximate integer common divisors. *International Conference on Cryptography and Lattices – Proceedings of CaLC '01*, Springer-Verlag, 2001, 51–66.

[6] Lenstra A. K., Lenstra H. W., Lovász L.: Factoring Polynomials with Rational Coefficients. Mathematische Ann. 261, Springer-Verlag, 1982, 515–534.

[7] Montgomery P. L.: Modular Multiplication without Trial Division. *Mathematics of Computation 44*, 1985, 519–512.

[8] Primas M.: Master thesis Kryptoanalýza s využitím postranní informace. 2010.

[9] RSA Data Security, Inc.: PKCS #1: RSA Encryption Standard. 2002.

[10] The OpenSSL Project: OpenSSL: The Open Source toolkit for SSL/TLS, 2003, `www.openssl.org`.

[11] Tomoeda Y., Miyake H., Shimbo A., aj.: An SPA-Based Extension of Schindler's Timing Attack against RSA Using CRT. *IEICE Transaction 88-A*, 2005, 147–153.

[12] Vaudenay S.: Personal communication in 2009.